Data Mining: Exploring OUA Basketball

Michael Armanious

05 June 2019

Contents

1 Introduction: Progress of the First Month

This month was mostly dedicated to data scraping from the OUA (Ontario University Athletics) website using Python and a Python library called BeautifulSoup which is used for pulling out data out of HTML and XML files. There was quite a learning curve since I had never done it before and it is quite a time consuming process.

In the end I was able to collect game by game data from the 2014/2015 Basketball Season for each Men's OUA Team until the most recent 2018/2019 Basketball Season. A Jupyter Notebook showing the code and the proper documentation is provided in the Appendix attached.

I then explored the data with R & SQL using visualizations and summary statistics functions. In addition I briefly tried a few classification and regression techniques.

2 Data Exploration

After scraping our desired data using Python and creating a dataframe, we need to explore the data to get a better picture of it before doing analysis. To do so one should always answer these two questions about the data: 1) What type of variation occurs within the variables? 2) What type of covariation occurs between the variables? First we need to read the file and see the dimensions of the dataframe.

```
library(reticulate)
df<-read.csv('C:/Basketball/gamebygame2.csv', header = T)
df<-as.data.frame(df)
dim(df)
```

```
## [1] 1171   59
```

```
library(DBI)
con <- dbConnect(RSQLite::SQLite(), ":memory:")
dbWriteTable(con, "gamebygame", df)
dbListFields(con, "gamebygame")
```

```
##  [1] "Game.ID"                   "X2nd.Chance.Points.Away"
##  [3] "X2nd.Chance.Points.Home"   "X3.Point...Away"
##  [5] "X3.Point...Home"           "X3PM.Away"
##  [7] "X3PA.Away"                 "X3PM.Home"
##  [9] "X3PA.Home"                 "Assists.Away"
## [11] "Assists.Home"              "Away"
## [13] "Bench.Points.Away"         "Bench.Points.Home"
## [15] "Fastbreak.Points.Away"     "Fastbreak.Points.Home"
## [17] "Field.Goal...Away"         "Field.Goal...Home"
## [19] "FGM.Away"                  "FGA.Away"
## [21] "FGM.Home"                  "FGA.Home"
## [23] "Free.Throw...Away"         "Free.Throw...Home"
## [25] "FTM.Away"                  "FTA.Away"
## [27] "FTM.Home"                  "FTA.Home"
## [29] "Home"                      "Largest.Lead.Away"
## [31] "Largest.Lead.Home"         "Lead.Changes"
## [33] "Loser"                     "Loser.1st.Qtr.Pts"
## [35] "Loser.2nd.Qtr.Pts"         "Loser.3rd.Qtr.Pts"
## [37] "Loser.4th.Qtr.Pts"         "Loser.OT.Points"
```

```
## [39] "Loser.Total.Pts"           "Points.in.the.Paint.Away"
## [41] "Points.in.the.Paint.Home"  "Points.off.Turnovers.Away"
## [43] "Points.off.Turnovers.Home" "Rebounds.Away"
## [45] "Rebounds.Home"             "Ties"
## [47] "Time.of.Largest.Lead.Away" "Time.of.Largest.Lead.Home"
## [49] "Turnovers.Away"            "Turnovers.Home"
## [51] "Winner"                    "Winner.1st.Qtr.Pts"
## [53] "Winner.2nd.Qtr.Pts"        "Winner.3rd.Qtr.Pts"
## [55] "Winner.4th.Qtr.Pts"        "Winner.OT.Points"
## [57] "Winner.Total.Pts"          "Season"
## [59] "Date"
```

Table 1: Features & Descriptions

Features	Description
Game ID	a unique ID to reference the games (row data)
Date	the date that the game took place
Season	the year of the regular season that the game took place
Home	Name of the team playing at Home
Away	Name of the team playing Away
FGM Away	Number of Field Goals Made for the AWay team
FGM Home	Number of Field Goals Made for the Home team
FGA Away	Number of Field Goals Attempted for the Away team
FGA Home	Number of Field Goals Attempted for the Home team
Field Goal% Away	(FGM/FGA) x 100 for Away team
Field Goal% Home	(FGM/FGA) x 100 for Home team
3PM Away	Number of 3 Pointers Made by the Away team
3PM Home	Number of 3 Pointers Made by the Home team
3PA Away	Number of 3 Pointers Attempted by the Away team
3PA Home	Number of 3 Pointers Attempted by the Away team
3Point% Away	(3PM/3PA) x 100 for Away team
3Point% Home	(3PM/3PA) x 100 for Home team
FTM Away	Number of FreeThrows Made by the Away team
FTM Home	Number of FreeThrows Made by the Home team
FTA Away	Number of FreeThrows Attempted by the Away team
FTA Home	Number of FreeThrows Attempted by the Home team
FT% Away	(FTM/FTA) x 100 for Away team
FT% Home	(FTM/FTA) x 100 for Home team
Assists Away	Number of Assists the Away team made
Assists Home	Number of Assists the Home team made
Rebounds Away	Number of Rebounds the Away team made
Rebounds Home	Number of Rebounds the Home team made
Turnovers Away	Number of Turnovers by the Away team
Turnovers Home	Number of Turnovers by the Home team
Points off Turnovers Away	Number of Points made off Turnovers by the Away team
Points off Turnovers Home	Number of Points made off Turnovers by the Home team
Points in the Paint Away	Number of Points made in the Paint by the Away team
Points in the Paint Home	Number of Points made in the Paint by the Home team
2nd Chance Points Away	Total 2nd Chance Pts for Away team
2nd Chance Points Home	Total 2nd Chance Pts for Home team
Bench Points Away	Number of Pts made by the bench players for the Away team
Bench Points Home	Number of Pts made by the bench players for the Home team
Fastbreak Pts Away	Number of Fastbreak Pts by the Away team

Features	Description
Fastbreak Pts Home	Number of Fastbreak Pts by the Home team
Largest Lead Away	The Largest Lead made by the Away team
Largest Lead Home	The Largest Lead made by the Home team
Time of Largest Lead Away	The time that the Away team had their Largest Lead
Time of Largest Lead Home	The time that the Home team had their Largest Lead
Winner	The name of the team that won the game
Winner 1st Qtr Pts	The number of points the winner scored in the 1st qtr
Winner 2nd Qtr Pts	The number of points the winner scored in the 2nd qtr
Winner 3rd Qtr Pts	The number of points the winner scored in the 3rd qtr
Winner 4th Qtr Pts	The number of points the winner scored in the 4th qtr
Winner OT Pts	The number of points the winner scored in overtime
Loser	The name od the team that lost the game
Loser 1st Qtr Pts	The number of points the loser scored in the 1st qtr
Loser 2nd Qtr Pts	The number of points the loser scored in the 2nd qtr
Loser 3rd Qtr Pts	The number of points the loser scored in the 3rd qtr
Loser 4th Qtr Pts	The number of points the loser scored in the 4th qtr
Loser OT Pts	The number of points the loser scored in overtime

We can see that there are 1171 rows of data and 59 columns/variables. This data consists of regular seasons starting from 2014-15 to the most recent season, 2018-19 for all the Men's Basketball Teams that are in the OUA (Ontario University Association) division. A table of the variables can be seen above with descriptions for each attribute.

```
head(df)
```

```
##                                                                                                  Game.ID
## 1 http://oua.ca/sports/mbkb/2018-19/boxscores/20180929_ket6.xml?view=teamstats
## 2 http://oua.ca/sports/mbkb/2015-16/boxscores/20151114_ml0v.xml?view=teamstats
## 3 http://oua.ca/sports/mbkb/2018-19/boxscores/20181201_7ufz.xml?view=teamstats
## 4 http://oua.ca/sports/mbkb/2015-16/boxscores/20151127_2xxv.xml?view=teamstats
## 5 http://oua.ca/sports/mbkb/2015-16/boxscores/20151107_pd4r.xml?view=teamstats
## 6 http://oua.ca/sports/mbkb/2016-17/boxscores/20161111_byiv.xml?view=teamstats
##   X2nd.Chance.Points.Away X2nd.Chance.Points.Home X3.Point...Away
## 1                      12                       7           0.500
## 2                      10                       7           0.368
## 3                      13                      10           0.353
## 4                      16                      21           0.346
## 5                      10                       9           0.133
## 6                      17                      22           0.269
##   X3.Point...Home X3PM.Away X3PA.Away X3PM.Home X3PA.Home Assists.Away
## 1           0.385        14        28        10        26           30
## 2           0.269         7        19         7        26            5
## 3           0.308        12        34         8        26           16
## 4           0.364         9        26         8        22            8
## 5           0.286         2        15         4        14           11
## 6           0.241         7        26         7        29           15
##   Assists.Home     Away Bench.Points.Away Bench.Points.Home
## 1           10 Carleton                59                29
## 2            6  Laurier                13                16
## 3           17  Queen's                18                17
## 4           12 Lakehead                 7                 8
```

```
## 5           10  Western                16                76
## 6           13 Lakehead                19                14
##   Fastbreak.Points.Away Fastbreak.Points.Home Field.Goal...Away
## 1                     0                    0             0.580
## 2                    NA                   NA             0.338
## 3                     5                    0             0.432
## 4                    NA                   NA             0.389
## 5                    NA                   NA             0.397
## 6                     6                    4             0.367
##   Field.Goal...Home FGM.Away FGA.Away FGM.Home FGA.Home Free.Throw...Away
## 1             0.393       40       69       24       61             0.765
## 2             0.309       27       80       21       68             0.692
## 3             0.371       32       74       26       70             0.571
## 4             0.379       28       72       33       87             0.625
## 5             0.347       27       68       25       72             0.750
## 6             0.375       29       79       30       80             0.524
##   Free.Throw...Home FTM.Away FTA.Away FTM.Home FTA.Home     Home
## 1             0.889       13       17        8        9 Bishop's
## 2             0.750       18       26       21       28   Algoma
## 3             0.690       12       21       20       29 McMaster
## 4             0.727       15       24        8       11    Brock
## 5             0.957       15       20       22       23  Laurier
## 6             0.789       11       21       15       19 McMaster
##   Largest.Lead.Away Largest.Lead.Home Lead.Changes    Loser
## 1                41                 0            0 Bishop's
## 2                12                 4            5   Algoma
## 3                 8                 4           13 McMaster
## 4                 5                 8           11 Lakehead
## 5                 9                 9           10  Western
## 6                 7                 8           11 Lakehead
##   Loser.1st.Qtr.Pts Loser.2nd.Qtr.Pts Loser.3rd.Qtr.Pts Loser.4th.Qtr.Pts
## 1                 7                20                23                16
## 2                14                14                18                22
## 3                18                18                19                23
## 4                18                17                23                18
## 5                19                19                16                12
## 6                 6                25                18                22
##   Loser.OT.Points Loser.Total.Pts Points.in.the.Paint.Away
## 1               0              66                        8
## 2               2              70                       NA
## 3               2              80                       34
## 4               4              80                       NA
## 5               5              71                       NA
## 6               5              76                       30
##   Points.in.the.Paint.Home Points.off.Turnovers.Away
## 1                        4                       23
## 2                       NA                       18
## 3                       32                       17
## 4                       NA                       11
## 5                       NA                       19
## 6                       36                        7
##   Points.off.Turnovers.Home Rebounds.Away Rebounds.Home Ties
## 1                         5            36            24    0
## 2                        13            49            41    5
```

```
## 3                       24            47            47   14
## 4                       18            47            55   14
## 5                       15            43            31   12
## 6                       17            44            44   11
##   Time.of.Largest.Lead.Away Time.of.Largest.Lead.Home Turnovers.Away
## 1              4th -- 00:28                       --              6
## 2              4th -- 08:00             1st -- 05:57             17
## 3               OT -- 00:11             1st -- 08:21             17
## 4              4th -- 09:24             2nd -- 03:56             14
## 5              2nd -- 01:48             1st -- 03:54             14
## 6              3rd -- 07:15             4th -- 05:07             14
##   Turnovers.Home   Winner Winner.1st.Qtr.Pts Winner.2nd.Qtr.Pts
## 1             20 Carleton                 29                 27
## 2             19  Laurier                 20                 13
## 3             18  Queen's                 18                 24
## 4             11    Brock                 20                 19
## 5             14  Laurier                 20                 11
## 6             15 McMaster                 13                 15
##   Winner.3rd.Qtr.Pts Winner.4th.Qtr.Pts Winner.OT.Points Winner.Total.Pts
## 1                 30                 21                0              107
## 2                 21                 14               11               79
## 3                 12                 24               10               88
## 4                 17                 20                6               82
## 5                 23                 12               10               76
## 6                 23                 20               11               82
##    Season     Date
## 1 2018-19 20180929
## 2 2015-16 20151114
## 3 2018-19 20181201
## 4 2015-16 20151127
## 5 2015-16 20151107
## 6 2016-17 20161111
```

This is a preview of some columns of the data. As we can see there are some missing entries. This is because some of the games did not collect certain fields. These fields are Fastbreak Points, Largest Lead, Time of Largest Lead, Points in the Paint, and Overtime Points.

2.1 Summary Statistics of Variables

Now we will look at summary statistics for every variable.

```
summary(df)
```

```
##                                                                  Game.ID
##  http://oua.ca/sports/mbkb/2014-15/boxscores/20141003_wswk.xml?view=teamstats:   1
##  http://oua.ca/sports/mbkb/2014-15/boxscores/20141004_i0t0.xml?view=teamstats:   1
##  http://oua.ca/sports/mbkb/2014-15/boxscores/20141004_wj8p.xml?view=teamstats:   1
##  http://oua.ca/sports/mbkb/2014-15/boxscores/20141005_kxa2.xml?view=teamstats:   1
##  http://oua.ca/sports/mbkb/2014-15/boxscores/20141005_ph9j.xml?view=teamstats:   1
##  http://oua.ca/sports/mbkb/2014-15/boxscores/20141015_zr3r.xml?view=teamstats:   1
##  (Other)                                                                    :1165
##  X2nd.Chance.Points.Away X2nd.Chance.Points.Home X3.Point...Away
```

```
##  Min.   : 0.0            Min.   : 0.00           Min.   :0.0000
##  1st Qu.: 6.0            1st Qu.: 7.00           1st Qu.:0.2500
##  Median :10.0            Median :10.00           Median :0.3160
##  Mean   :10.3            Mean   :10.87           Mean   :0.3214
##  3rd Qu.:14.0            3rd Qu.:14.00           3rd Qu.:0.3910
##  Max.   :31.0            Max.   :32.00           Max.   :1.0000
##  NA's   :3               NA's   :3
##  X3.Point...Home    X3PM.Away        X3PA.Away       X3PM.Home
##  Min.   :0.0000   Min.   : 0.000   Min.   : 0.00   Min.   : 0.00
##  1st Qu.:0.2630   1st Qu.: 6.000   1st Qu.:19.00   1st Qu.: 6.00
##  Median :0.3330   Median : 7.000   Median :24.00   Median : 8.00
##  Mean   :0.3304   Mean   : 7.687   Mean   :23.96   Mean   : 8.16
##  3rd Qu.:0.4000   3rd Qu.: 9.000   3rd Qu.:28.00   3rd Qu.:10.00
##  Max.   :1.0000   Max.   :20.000   Max.   :56.00   Max.   :23.00
##
##    X3PA.Home      Assists.Away    Assists.Home         Away
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Ottawa  : 69
##  1st Qu.:20.00   1st Qu.: 8.00   1st Qu.:11.00   Algoma  : 65
##  Median :24.00   Median :11.00   Median :14.00   Western : 63
##  Mean   :24.48   Mean   :11.97   Mean   :14.38   Guelph  : 62
##  3rd Qu.:29.00   3rd Qu.:15.00   3rd Qu.:18.00   Laurier : 62
##  Max.   :52.00   Max.   :32.00   Max.   :33.00   Carleton: 61
##                                                  (Other) :789
##  Bench.Points.Away Bench.Points.Home Fastbreak.Points.Away
##  Min.   : 0.0      Min.   :  0.00    Min.   : 0.000
##  1st Qu.:15.0      1st Qu.: 16.00    1st Qu.: 0.000
##  Median :23.0      Median : 23.00    Median : 0.000
##  Mean   :25.3      Mean   : 25.96    Mean   : 2.211
##  3rd Qu.:31.0      3rd Qu.: 32.00    3rd Qu.: 3.000
##  Max.   :99.0      Max.   :110.00    Max.   :28.000
##  NA's   :3         NA's   :3         NA's   :792
##  Fastbreak.Points.Home Field.Goal...Away Field.Goal...Home    FGM.Away
##  Min.   : 0.000        Min.   :0.0000    Min.   :0.0000    Min.   : 0.00
##  1st Qu.: 0.000        1st Qu.:0.3600    1st Qu.:0.3730    1st Qu.:24.00
##  Median : 0.000        Median :0.4110    Median :0.4260    Median :27.00
##  Mean   : 2.715        Mean   :0.4128    Mean   :0.4252    Mean   :27.43
##  3rd Qu.: 4.000        3rd Qu.:0.4620    3rd Qu.:0.4770    3rd Qu.:31.00
##  Max.   :25.000        Max.   :0.6620    Max.   :0.6720    Max.   :51.00
##  NA's   :792
##     FGA.Away         FGM.Home        FGA.Home      Free.Throw...Away
##  Min.   :  0.00   Min.   : 0.00   Min.   :  0.00   Min.   :0.0000
##  1st Qu.: 61.00   1st Qu.:25.00   1st Qu.: 61.00   1st Qu.:0.6110
##  Median : 66.00   Median :28.00   Median : 67.00   Median :0.6960
##  Mean   : 66.48   Mean   :28.43   Mean   : 66.88   Mean   :0.6894
##  3rd Qu.: 72.00   3rd Qu.:32.00   3rd Qu.: 72.00   3rd Qu.:0.7780
##  Max.   :102.00   Max.   :47.00   Max.   :100.00   Max.   :1.0000
##
##  Free.Throw...Home    FTM.Away        FTA.Away        FTM.Home
##  Min.   :0.0000    Min.   : 0.00   Min.   : 0.00   Min.   : 0.0
##  1st Qu.:0.6150    1st Qu.: 9.00   1st Qu.:14.00   1st Qu.:10.0
##  Median :0.6970    Median :13.00   Median :19.00   Median :13.0
##  Mean   :0.6938    Mean   :13.41   Mean   :19.43   Mean   :13.7
##  3rd Qu.:0.7830    3rd Qu.:17.00   3rd Qu.:24.00   3rd Qu.:17.0
##  Max.   :1.0000    Max.   :38.00   Max.   :52.00   Max.   :35.0
```

```
##
##     FTA.Home           Home     Largest.Lead.Away Largest.Lead.Home
##  Min.   : 0.00   Carleton: 72   Min.   : 0.00     Min.   : 0.00
##  1st Qu.:14.00   Lakehead: 72   1st Qu.: 4.00     1st Qu.: 5.00
##  Median :19.00   Ottawa  : 72   Median : 9.00     Median :11.00
##  Mean   :19.65   Queen's : 70   Mean   :11.23     Mean   :13.43
##  3rd Qu.:24.00   Toronto : 70   3rd Qu.:16.00     3rd Qu.:19.00
##  Max.   :50.00   McMaster: 65   Max.   :62.00     Max.   :70.00
##                  (Other) :750   NA's   :222       NA's   :222
##   Lead.Changes          Loser     Loser.1st.Qtr.Pts Loser.2nd.Qtr.Pts
##  Min.   : 0.000   Waterloo : 94   Min.   : 2.00     Min.   : 0.0
##  1st Qu.: 1.000   Algoma   : 91   1st Qu.:13.00     1st Qu.:13.0
##  Median : 3.000   Nipissing: 88   Median :16.00     Median :16.0
##  Mean   : 4.149   York     : 84   Mean   :16.22     Mean   :16.5
##  3rd Qu.: 7.000   Lakehead : 81   3rd Qu.:20.00     3rd Qu.:20.0
##  Max.   :23.000   Guelph   : 78   Max.   :33.00     Max.   :34.0
##  NA's   :8        (Other)  :655
##  Loser.3rd.Qtr.Pts Loser.4th.Qtr.Pts Loser.OT.Points  Loser.Total.Pts
##  Min.   : 2.00     Min.   : 4.00     Min.   : 0.000   Min.   : 36.0
##  1st Qu.:14.00     1st Qu.:15.00     1st Qu.: 7.000   1st Qu.: 61.0
##  Median :18.00     Median :18.00     Median : 8.000   Median : 69.0
##  Mean   :17.98     Mean   :18.31     Mean   : 8.405   Mean   : 69.3
##  3rd Qu.:21.00     3rd Qu.:22.00     3rd Qu.:10.000   3rd Qu.: 77.0
##  Max.   :38.00     Max.   :39.00     Max.   :17.000   Max.   :110.0
##                    NA's   :1         NA's   :1129
##  Points.in.the.Paint.Away Points.in.the.Paint.Home
##  Min.   : 0.00            Min.   : 0.00
##  1st Qu.: 0.00            1st Qu.: 2.00
##  Median :20.00            Median :24.00
##  Mean   :18.72            Mean   :21.47
##  3rd Qu.:32.00            3rd Qu.:36.00
##  Max.   :62.00            Max.   :62.00
##  NA's   :792              NA's   :792
##  Points.off.Turnovers.Away Points.off.Turnovers.Home Rebounds.Away
##  Min.   : 0.00             Min.   : 0.00             Min.   : 0.00
##  1st Qu.:10.00             1st Qu.:10.00             1st Qu.:32.00
##  Median :13.00             Median :14.00             Median :37.00
##  Mean   :14.31             Mean   :15.32             Mean   :37.29
##  3rd Qu.:18.00             3rd Qu.:20.00             3rd Qu.:43.00
##  Max.   :45.00             Max.   :42.00             Max.   :64.00
##  NA's   :3                 NA's   :3
##  Rebounds.Home        Ties        Time.of.Largest.Lead.Away
##  Min.   : 0.00   Min.   : 0.000              :222
##  1st Qu.:33.00   1st Qu.: 1.000    --        : 31
##  Median :38.00   Median : 2.000   4th -- 00:00:  7
##  Mean   :38.32   Mean   : 3.187   4th -- 00:05:  5
##  3rd Qu.:44.00   3rd Qu.: 5.000   1st -- 08:33:  4
##  Max.   :75.00   Max.   :21.000   1st -- 09:16:  4
##                  NA's   :8        (Other)     :898
##  Time.of.Largest.Lead.Home Turnovers.Away Turnovers.Home      Winner
##              :222          Min.   : 0.0   Min.   : 0.0   Carleton:122
##   --         : 21          1st Qu.:11.0   1st Qu.:11.0   Ottawa  :116
##  4th -- 00:00:  6          Median :14.0   Median :14.0   Ryerson : 95
##  4th -- 00:34:  6          Mean   :14.6   Mean   :14.2   Brock   : 76
```

```
##  1st -- 08:34:  4          3rd Qu.:18.0   3rd Qu.:17.0   McMaster: 75
##  4th -- 00:15:  4          Max.   :33.0   Max.   :31.0   Western : 66
##  (Other)     :908                                       (Other) :621
##  Winner.1st.Qtr.Pts Winner.2nd.Qtr.Pts Winner.3rd.Qtr.Pts
##  Min.   : 3.00      Min.   : 2.00      Min.   : 5.00
##  1st Qu.:17.00      1st Qu.:17.00      1st Qu.:18.00
##  Median :21.00      Median :20.00      Median :22.00
##  Mean   :20.81      Mean   :20.56      Mean   :21.97
##  3rd Qu.:24.00      3rd Qu.:24.00      3rd Qu.:26.00
##  Max.   :49.00      Max.   :38.00      Max.   :39.00
##
##  Winner.4th.Qtr.Pts Winner.OT.Points Winner.Total.Pts     Season
##  Min.   : 6.00      Min.   : 0.00    Min.   : 52.00   2014-15:188
##  1st Qu.:18.00      1st Qu.:10.00    1st Qu.: 78.00   2015-16:219
##  Median :22.00      Median :13.00    Median : 85.00   2016-17:221
##  Mean   :21.71      Mean   :12.83    Mean   : 85.49   2017-18:276
##  3rd Qu.:25.00      3rd Qu.:15.75    3rd Qu.: 92.50   2018-19:267
##  Max.   :40.00      Max.   :21.00    Max.   :139.00
##  NA's   :1          NA's   :1129
##       Date
##  Min.   :20141003
##  1st Qu.:20151128
##  Median :20170128
##  Mean   :20167126
##  3rd Qu.:20180209
##  Max.   :20190216
##
```

From here we can see the average statistics along with the quartiles, max and min of the variables. Next we will do some visuals to see the distributions of variables.

2.2 Distribution/Variation of Variables

```
par(mfrow=c(3,3))
hist(df$X3.Point...Away, main = '3 Point% Away', xlab = '3FG%')
hist(df$X3.Point...Home, main = '3 Point% Home', xlab = '3FG%')
hist(df$Field.Goal...Away, main = 'Field Goal% Away', xlab = 'FG%')
hist(df$Field.Goal...Home, main = 'Field Goal% Home', xlab = 'FG%')
hist(df$Free.Throw...Away, main = 'FreeThrow% Away', xlab = 'FT%')
hist(df$Free.Throw...Home, main = 'FreeThrow% Home', xlab = 'FT%')
hist(df$Turnovers.Away, main = 'Turnovers Away', xlab = 'Turnovers')
hist(df$Turnovers.Home, main = 'Turnovers Home', xlab = 'Turnovers')
hist(df$Lead.Changes, main = 'Lead Changes', xlab = 'Amount of Lead Changes in a game' )
```

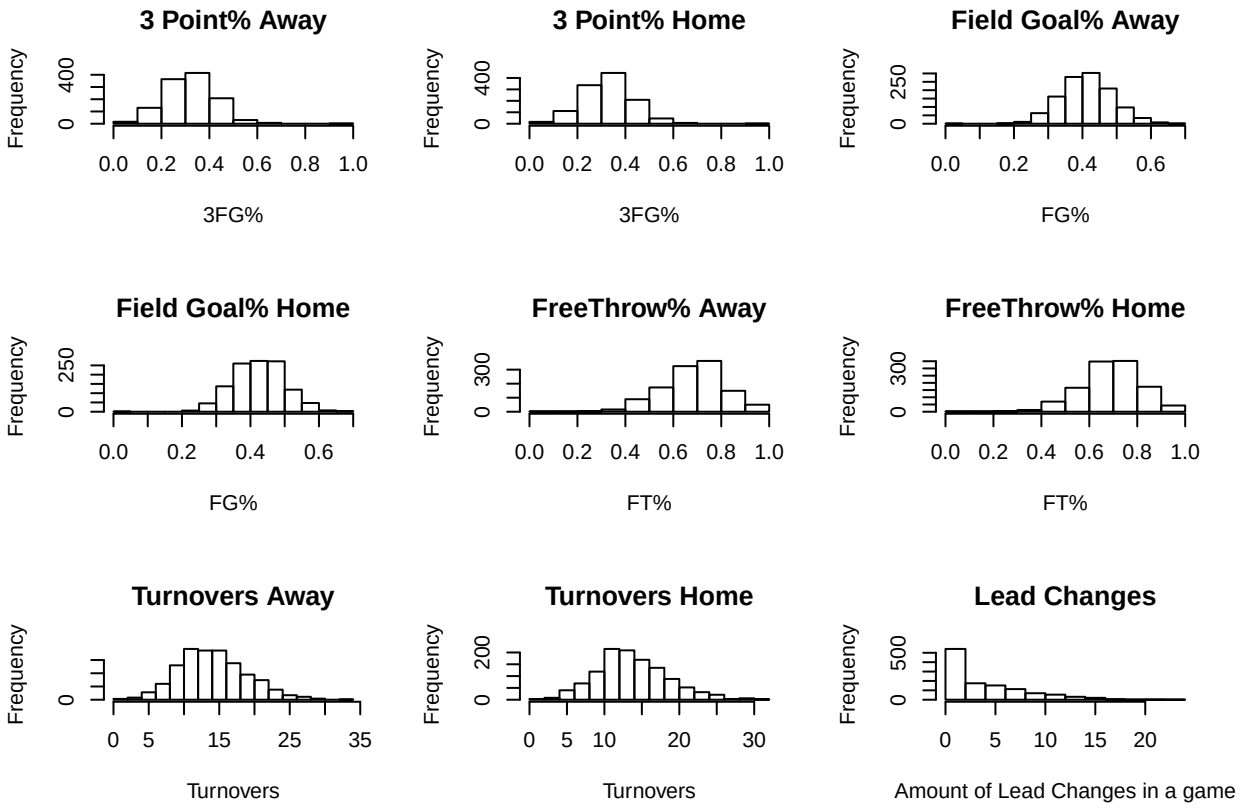


Figure 1: Distribution of Variables

Let's take a closer look at the difference of Field Goal % between Away and Home teams.

```
#Away FG%
summary(df$Field.Goal...Away)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.3600  0.4110  0.4128  0.4620  0.6620
```

```
#Home FG%
summary(df$Field.Goal...Home)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.3730  0.4260  0.4252  0.4770  0.6720
```

```
#Away 3FG%
summary(df$X3.Point...Away)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.2500  0.3160  0.3214  0.3910  1.0000
```

```r
#Home 3FG%
summary(df$X3.Point...Home)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.2630  0.3330  0.3304  0.4000  1.0000
```

Here we can see that when teams played at home, they have slightly better shooting efficiency. Now let's see if Teams won more when playing at Home than when they were playing Away to see if there is a Home Team advantage.

```r
library(DBI)
db = dbConnect(RSQLite::SQLite(), dbname = "sql.sqlite")
```

```r
SELECT SUM(CASE WHEN Winner=Home THEN 1 ELSE 0 END) AS HomeWin,
SUM(CASE WHEN Winner=Away THEN 1 ELSE 0 END) AS AwayWin
FROM gamebygame
```

Table 2: Home vs. Away Win Frequency

HomeWin	AwayWin
651	520

There is a significant difference between the amount of home wins and the amount of away wins.

Let's look at the winning and losing frequencies of every team.

```r
library(ggplot2)
theme_set(theme_bw())

dfwins<-as.data.frame(table(df$Winner))

g<-ggplot(dfwins, aes(x=reorder(Var1, -Freq), y=Freq))
  g + geom_bar(stat="identity", width=.5, fill="tomato3") +
  labs(title="Win Frequency",
       subtitle="Total Wins Per Team from 2014-15 to 2018-19") +
  theme(axis.text.x = element_text(angle=90, vjust=0.3)) +
    xlab('Teams') +
  ylab('Total Wins')
```

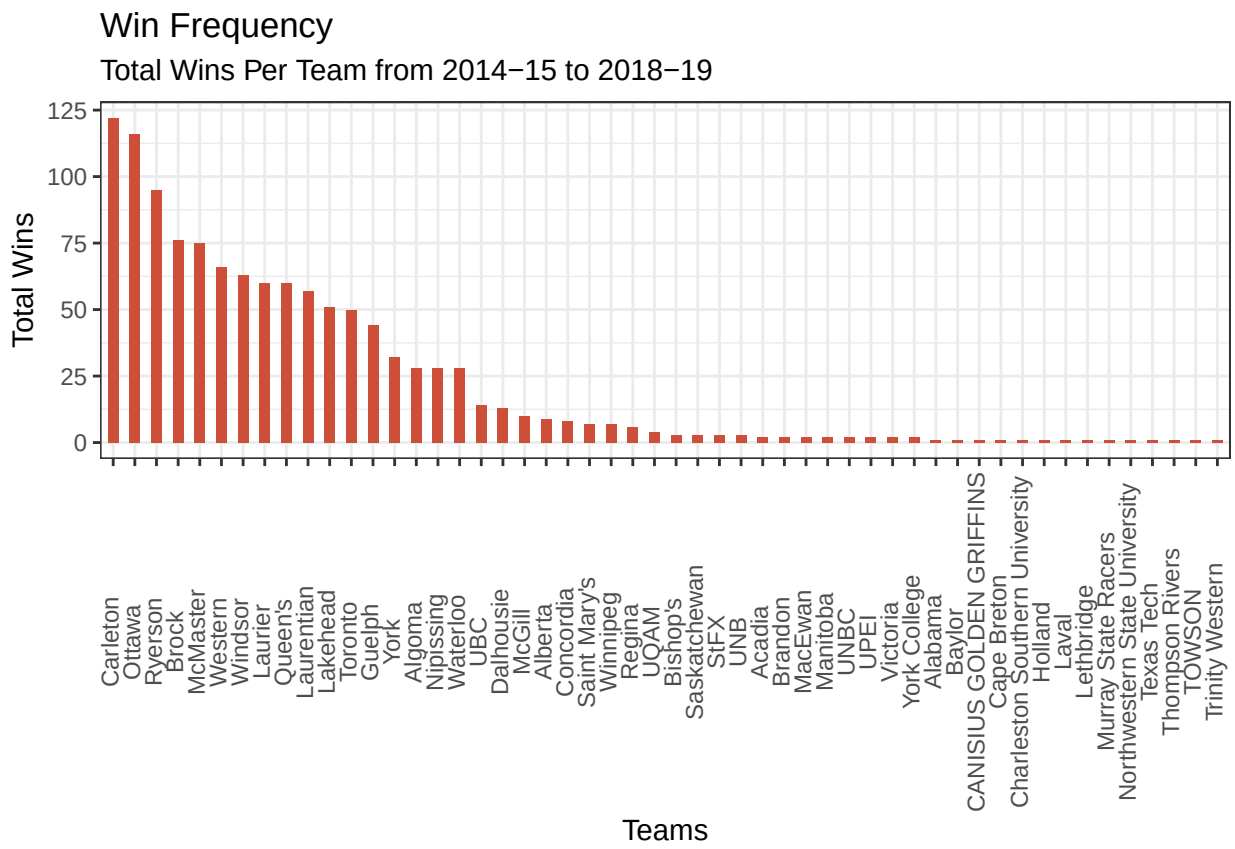


Figure 2: Winning Frequency

```
dfloss<-as.data.frame(table(df$Loser))

j<-ggplot(dfloss, aes(x=reorder(Var1, -Freq), y=Freq))
j + geom_bar(stat="identity", width=.5, fill="tomato3") +
  labs(title="Loss Frequency",
       subtitle="Total Losses Per Team from 2014-15 to 2018-19") +
  theme(axis.text.x = element_text(angle=90, vjust=0.3)) +
  xlab('Teams') +
  ylab('Total Losses')
```

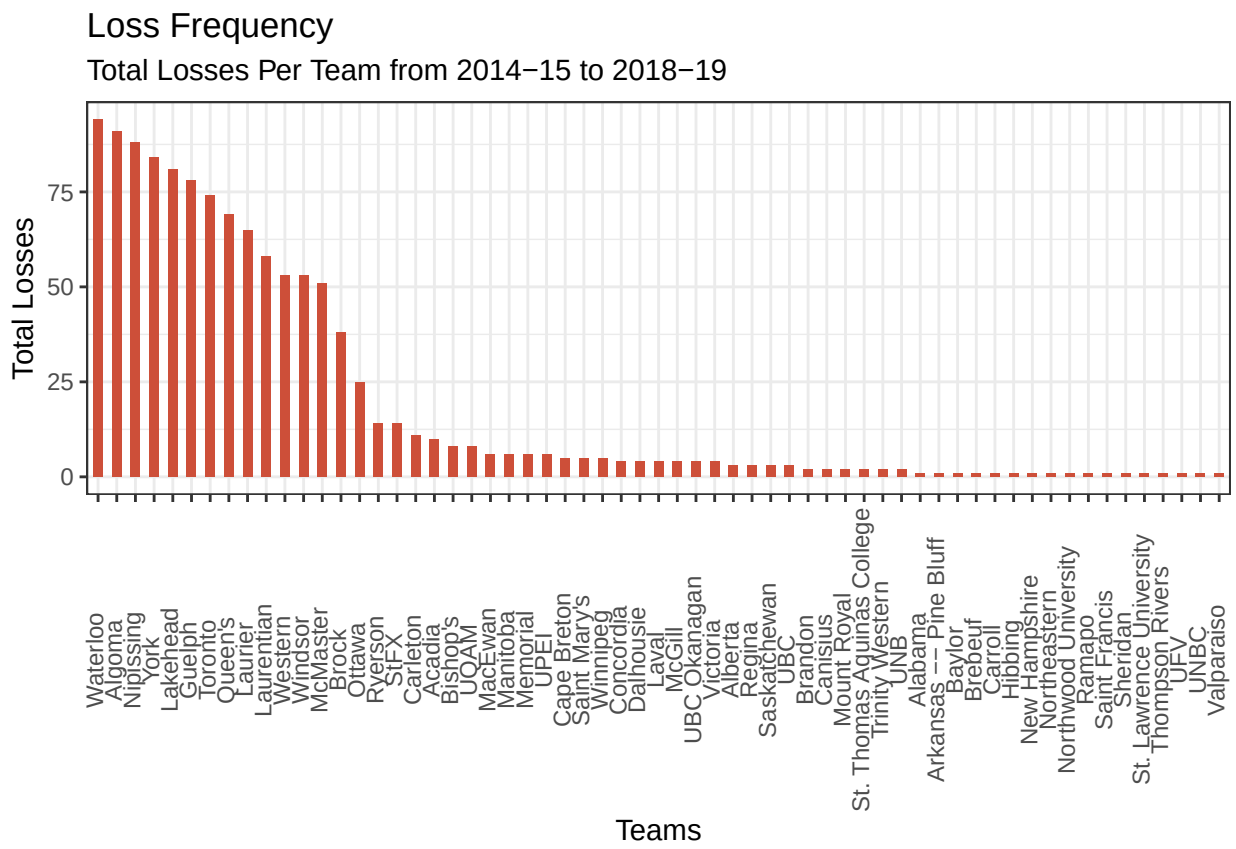


Figure 3: Losing Frequency

We can see that Carleton and Ottawa have the most wins from season 2014-15 to 2018-19. We can also see that there are long tails since some of these games were exhibition games against teams that are not a part of the OUA division. Note: Every team to the right of Waterloo in Figure 1 are not in the OUA division.

For further exploration we can stack the dataframe so that we do not need two of each variable (i.e making a row for each team rather than each row having data of two teams) and create binary variables for Win, and Home (i.e. if the team won then the variable is set to 1 and 0 otherwise. Likewise if the team is playing at home then the variable is set to 1 and 0 otherwise). This would create double the amount of rows.

```
df2<-read.csv(file='C:/Basketball/team data.csv', header = T)
df2<-as.data.frame(df2)
dim(df2)
```

```
## [1] 2343   31
```

```
head(df2)
```

```
##                                                                GameID
## 1 http://oua.ca/sports/mbkb/2014-15/boxscores/20141003_wswk.xml?view=teamstats
## 2 http://oua.ca/sports/mbkb/2014-15/boxscores/20141003_wswk.xml?view=teamstats
## 3 http://oua.ca/sports/mbkb/2014-15/boxscores/20141004_i0t0.xml?view=teamstats
## 4 http://oua.ca/sports/mbkb/2014-15/boxscores/20141004_i0t0.xml?view=teamstats
```

```
## 5 http://oua.ca/sports/mbkb/2014-15/boxscores/20141004_wj8p.xml?view=teamstats
## 6 http://oua.ca/sports/mbkb/2014-15/boxscores/20141004_wj8p.xml?view=teamstats
##              Team  Season   FT. FTM FTA   FG. FGM FGA X3Point. X3PM X3PA
## 1          Guelph 2014-15 0.581  18  31 0.338  24  71    0.321    9   28
## 2         Queen's 2014-15 0.580  29  50 0.451  23  51    0.250    3   12
## 3         Queen's 2014-15 0.731  19  26 0.353  18  51    0.227    5   22
## 4 Trinity Western 2014-15 0.563  18  32 0.400  24  60    0.167    3   18
## 5       Dalhousie 2014-15 0.704  19  27 0.509  29  57    0.364    8   22
## 6         Toronto 2014-15 0.833  20  24 0.369  24  65    0.321    9   28
##   SecondChancePts Assists Rebounds BenchPoints FastbreakPts
## 1              20       8       40          30           NA
## 2               7       8       37          19           NA
## 3               2       7       34          15           NA
## 4               9      11       46          16           NA
## 5              15      17       42          29            0
## 6              16      16       37           9            0
##   PointsinthePaint PointsOffTurnovers Turnovers LargestLead
## 1               NA                 11        15           7
## 2               NA                 12        13           8
## 3               NA                 19        30           7
## 4               NA                 32        28           9
## 5                0                 19        22          15
## 6                0                 17        17          10
##   TimeofLargestLead LeadChanges Ties FirstQtr SecondQtr ThirdQtr FourthQtr
## 1      1st -- 02:02          12    8       22        18       17        18
## 2      4th -- 07:17          12    8       18        26       18        16
## 3      1st -- 06:29           5    8       12        16       20        12
## 4      2nd -- 09:01           5    8       18        15       15        21
## 5      4th -- 03:59           7    7       17        22       24        22
## 6      1st -- 03:00           7    7       24        16       16        21
##   TotalPoints Win Home
## 1          75   0    0
## 2          78   1    1
## 3          60   0    1
## 4          69   1    0
## 5          85   1    0
## 6          77   0    1
```

2.3 Correlations

Now we'll take a look at the correlations between all the numeric variables.

```
df2.sub<-df2[,c(4:21,23:29)]
library(ggcorrplot)

# Correlation matrix
corr <- round(cor(df2.sub,use = "pairwise.complete.obs"), 3)

# Plot
ggcorrplot(corr, hc.order = TRUE,
           type = "lower",
           lab = TRUE,
           lab_size = 3,
```

```
          method="circle",
          colors = c("tomato2", "white", "springgreen3"),
          title="Correlogram of Team Data",
          ggtheme=theme_bw)
```

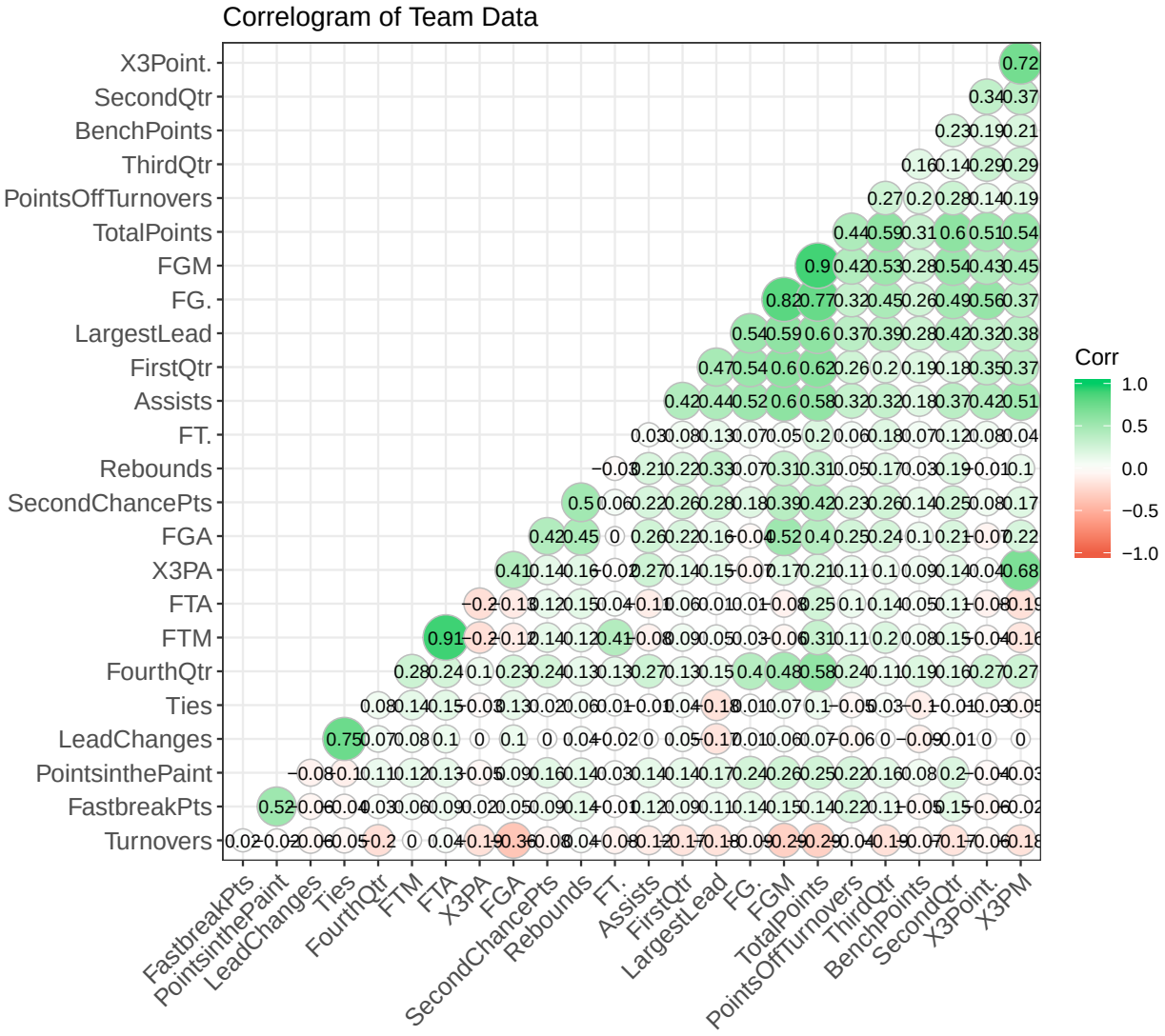


Figure 4: Correlogram of Features

Here we can see that the total points are highly correlated to the field-Goals made obviously and that the only variable that is negatively correlated to total points is turnovers. What's surprising is that freethrows are not highly correlated to any other variables. Ties are not highly correlated to anything as well. Rebounds are correlated with second chance points which makes sense but not as highly correlated to anything else, including points which is interesting.

```
library(GGally)
df2.sub<-df2[,c(4,7,10,13:21,23,24,29)]
ggpairs(df2.sub)
```

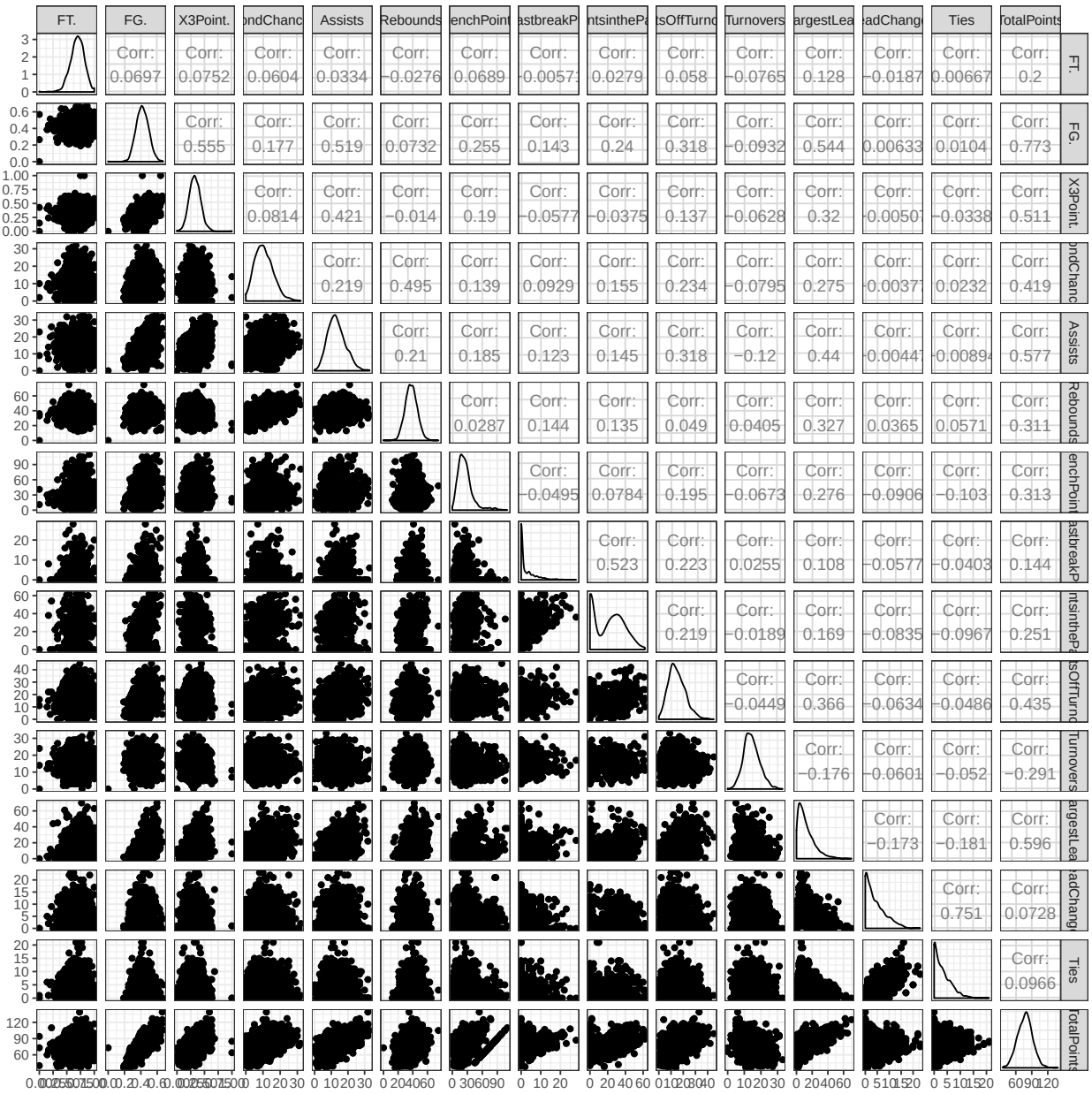


Figure 5: Distribution and Correlations of Pairs of Variables

This shows us scatterplots of different pairs of variables along with the associated correlation measurement. There aren't that many clear relationships between the pairs of variables.

3 Regression

After performing the exploratory procedures we can now move on to predictive analysis. First, we'll try different regression models. Regression analysis is a powerful statistical method that allows you to examine the relationship between two or more variables of interest. Regression analysis is for studying how independent predictors can affect the dependent variable (the dependent variable is the variable we are trying to predict through the use of the independent predictors). We will perform different kinds of regression and evaluate different models. First let's try using regression to predict the total points of a team. Our dependent variable will be points and we will try different combinations of independent variables and evaluate them to see which model provides the best fit.

3.1 Building the Linear Model

Typically for predicting linear models, we should make a training and test set but we will start by focusing on building a sound model.

3.2 First Model

First we will try a very basic linear model where,

$$TotalPoints = \beta_0 + \beta * FG$$

```
df3.sub<-df2[,c(4,7,10,13:21,23,24,29)]
linmod<-lm(TotalPoints ~ FG. + X3Point. + Assists + Rebounds + Turnovers, df3.sub)
firstlm<-lm(TotalPoints ~ FG., df3.sub)
summary(firstlm)
```

```
##
## Call:
## lm(formula = TotalPoints ~ FG., data = df3.sub)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.073  -5.960  -0.341   5.899  55.379
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   17.621      1.031   17.09   <2e-16 ***
## FG.          142.566      2.420   58.91   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.821 on 2341 degrees of freedom
## Multiple R-squared:  0.5972, Adjusted R-squared:  0.597
## F-statistic:  3470 on 1 and 2341 DF,  p-value: < 2.2e-16
```

```
AIC(firstlm)
```

```
## [1] 16855.41
```

```r
BIC(firstlm)
```

```
## [1] 16872.69
```

There are many different ways to evaluate regression models.

Table 3: Useful Measurements to Evaluate Regression Model

Statistic	Criterion
R-Squared	Higher the better (> 0.70)
Adj R-Squared	Higher the better
F-Statistic	Higher the better
Std. error	Closer to zero the better
t-statistic	Should be greater 1.96 for p-value to be less than 0.05
AIC	Lower the better
BIC	Lower the better
MSE	Lower the better

In this model we have $R^2 = 0.5972$ and Adj. R-Squared $= 0.597$. R-Squared tells us the proportion of variation in the dependent (response) variable that has been explained by this model.

Right off the bat we know that this model is not accurate at all. We also have a very high AIC and BIC.

3.3 Second Model

Now we'll try adding more independent variables that may be useful in creating a regression model.

$$TotalPoints = \beta_0 + \beta_1 * FG + \beta_2 * 3P + \beta_3 * FT + \beta_4 * Assists + \beta_5 * Rebounds + \beta_6 * Turnovers$$

```r
secondlm<-lm(TotalPoints ~ FG. + X3Point. + FT. + Assists + Rebounds + Turnovers, df3.sub)
summary(secondlm)
```

```
##
## Call:
## lm(formula = TotalPoints ~ FG. + X3Point. + FT. + Assists + Rebounds +
##     Turnovers, data = df3.sub)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.218  -4.438  -0.162   4.204  68.671
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.32905    1.34627   3.216  0.00132 **
## FG.         109.80605    2.42513  45.278  < 2e-16 ***
## X3Point.     13.35797    1.64922   8.100 8.79e-16 ***
## FT.          14.80225    1.10038  13.452  < 2e-16 ***
## Assists       0.37268    0.03249  11.471  < 2e-16 ***
## Rebounds      0.43298    0.01796  24.112  < 2e-16 ***
```

```
## Turnovers    -0.61516    0.02969 -20.716  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.8 on 2336 degrees of freedom
## Multiple R-squared:  0.7611, Adjusted R-squared:  0.7605
## F-statistic:  1240 on 6 and 2336 DF,  p-value: < 2.2e-16
```

```
AIC(secondlm)
```

```
## [1] 15641.15
```

```
BIC(secondlm)
```

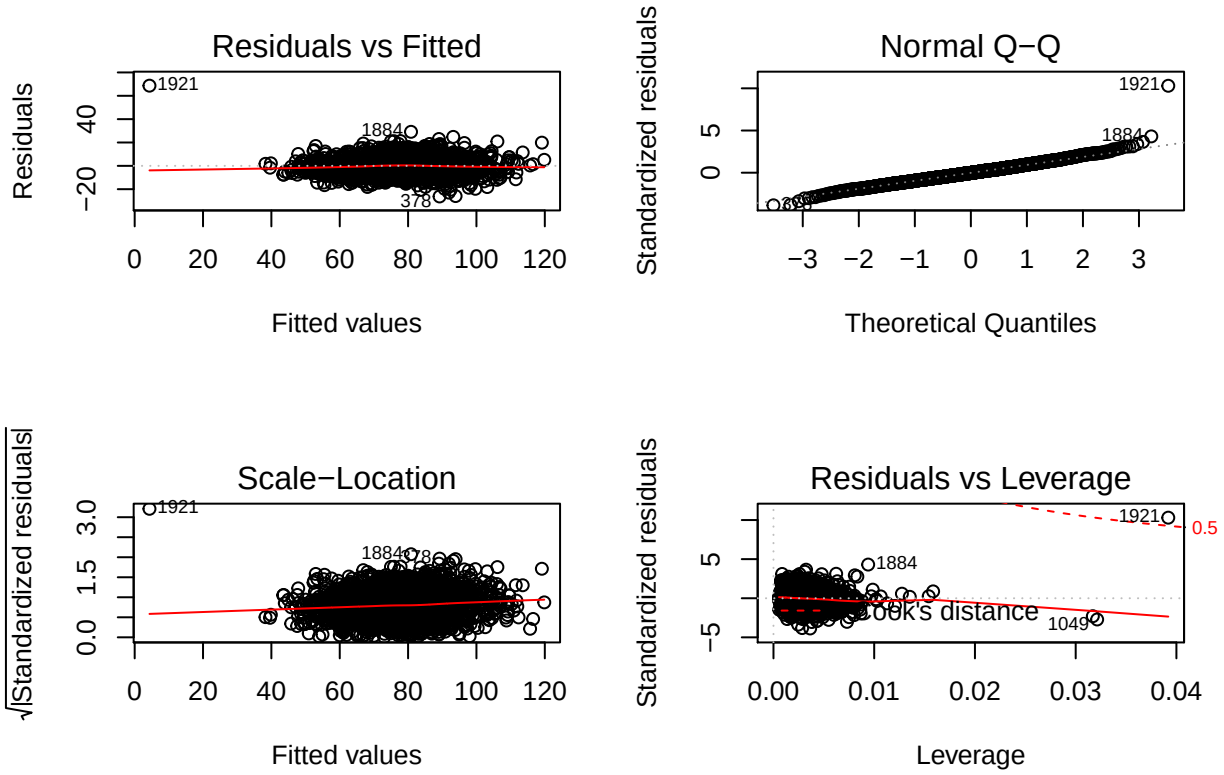
```
## [1] 15687.23
```

This model has an improved R^2 of 0.7611 and Adj. R-Squared of 0.7605. It has a lower F-statistic than our first model and lower AIC and BIC than the first model. All of the variables are very significant according to the p-values. However, overall this model is still not very good.

3.3.1 Diagnostics

There are many ways to assess a regression model.

```
par(mfrow=c(2,2))
plot(secondlm)
```



The first plot is a plot of the residuals vs the fitted (estimated responses).The plot is used to detect non-linearity, unequal variances and outliers. We can see that there is one outlier which is a game on October 20, 2018, York vs. Cape Breton where the team stats are all (http://oua.ca/sports/mbkb/2018-19/boxscores/20181020_7xwg.xml?view=teamstats) so we will delete the outlier and run the regression but this time with train and test sets.

```
d<-df2[-(df2$GameID == 'http://oua.ca/sports/mbkb/2018-19/boxscores/20181020_7xwg.xml?view=teamstats'),]

d1<-d[-c(1920),]

thirdlm<-lm(TotalPoints ~ FG. + X3Point. + FT. + Assists + Rebounds + Turnovers, d1)
summary(thirdlm)
```

```
##
## Call:
## lm(formula = TotalPoints ~ FG. + X3Point. + FT. + Assists + Rebounds +
##     Turnovers, data = d1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.561  -4.364  -0.124   4.165  29.833
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.48934    1.34209   1.110    0.267
## FG.         112.16435    2.37966  47.135   <2e-16 ***
## X3Point.     13.43681    1.61154   8.338   <2e-16 ***
## FT.          16.11383    1.08208  14.892   <2e-16 ***
## Assists       0.36575    0.03175  11.519   <2e-16 ***
## Rebounds      0.44991    0.01762  25.534   <2e-16 ***
## Turnovers    -0.59195    0.02909 -20.346   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.644 on 2334 degrees of freedom
## Multiple R-squared:  0.7722, Adjusted R-squared:  0.7716
## F-statistic:  1318 on 6 and 2334 DF,  p-value: < 2.2e-16
```

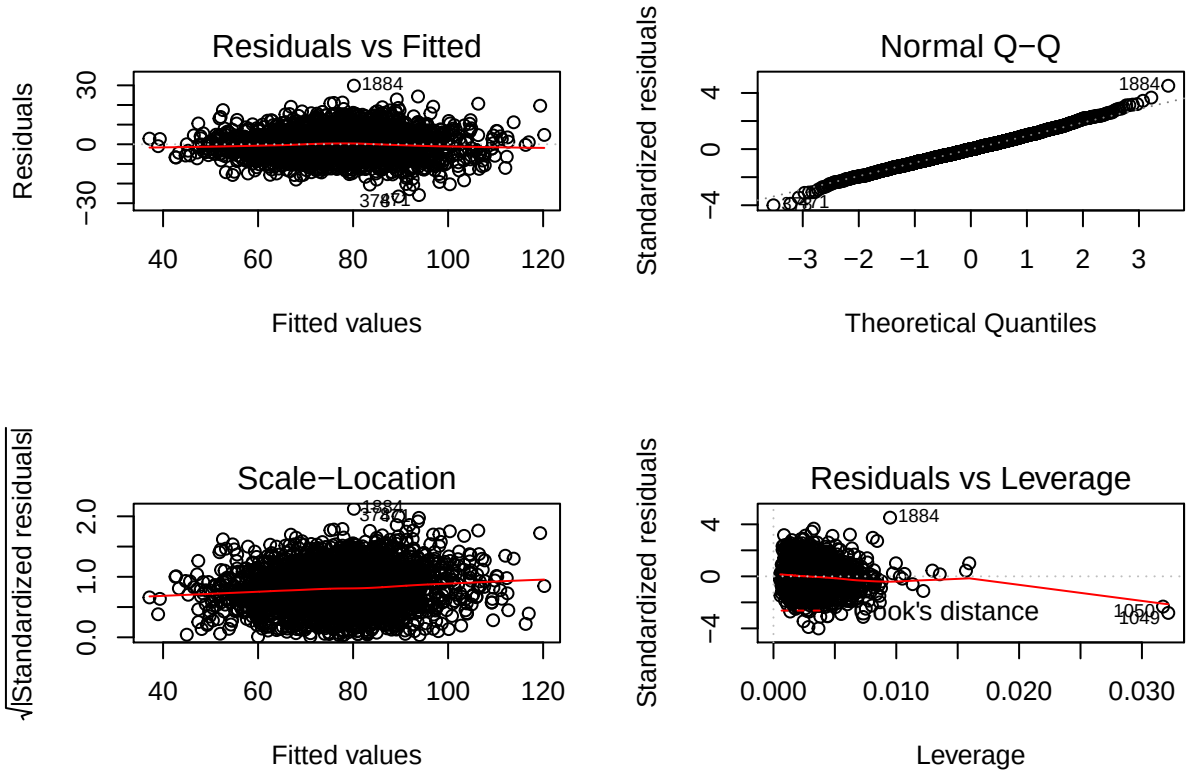
```
AIC(thirdlm)
```

```
## [1] 15518.75
```

```
BIC(thirdlm)
```

```
## [1] 15564.82
```

```
par(mfrow=c(2,2))
plot(thirdlm)
```



Now we can see there is no outlier and the QQ plot does not have any curvature to indicate non-normality.

3.4 Prediction

The training set will consist of 75% of the dataset and the test set will be the remaining 25%.

4 Risk Ratios and Odd Ratios

4.1 What are Risk Ratios and Odd Ratios

Risk Ratio (RR) or Relative Risk is a measurement often used in epidemiology. It is used to estimate the outcome between factors and outcomes. In our case we will use this measurement to see whether there is a statistically significant difference between teams playing at home versus away. A risk ratio of 1 means there is no difference, greater than 1 means there is a higher chance of winning if the team is playing at home and less than 1 means the opposite. An Odds Ratio (OR) is a ratio of ratios. It also quantifies the strength of the association between two events. If the odds ratio equals 1 then the odds of the events are the same. If the odds ratio is greater than 1 then the events are correlated in the sense that if compared to the absence of the second event, the presence of the second raises the odds of the first event, and symmetrically the presence of the first event raises the odds of the second event. In our case we will obtain both measurements to see the strength of association between teams playing at home versus teams playing away.

4.2 Results

```
library(Epi)
c.table<-array(data = c(651,520,520,651), dim = c(2,2),
          dimnames = list(Group = c("Away", "Home"),
                          Outcome = c("Lose", "Win")))
twoby2(c.table, alpha = 0.05)
```

```
## 2 by 2 table analysis:
## ------------------------------------------------------
## Outcome   : Lose
## Comparing : Away vs. Home
##
##      Lose Win    P(Lose) 95% conf. interval
## Away  651 520     0.5559    0.5273   0.5842
## Home  520 651     0.4441    0.4158   0.4727
##
##                                    95% conf. interval
##              Relative Risk: 1.2519    1.1533   1.3589
##          Sample Odds Ratio: 1.5673    1.3315   1.8448
## Conditional MLE Odds Ratio: 1.5670    1.3270   1.8512
##     Probability difference: 0.1119    0.0714   0.1518
##
##               Exact P-value: 0
##          Asymptotic P-value: 0
## ------------------------------------------------------
```

These results show that we are 95% confident that the odds of winning when playing at home (or equivalently losing while away) is 33%-84% compared to when a team wins while playing away. The risk ratio tells us that we are 95% confident that the 'risk' of a team losing when playing away is between 15% to 36% higher

5 Logistic Regression

5.1 What is Logistic Regression?

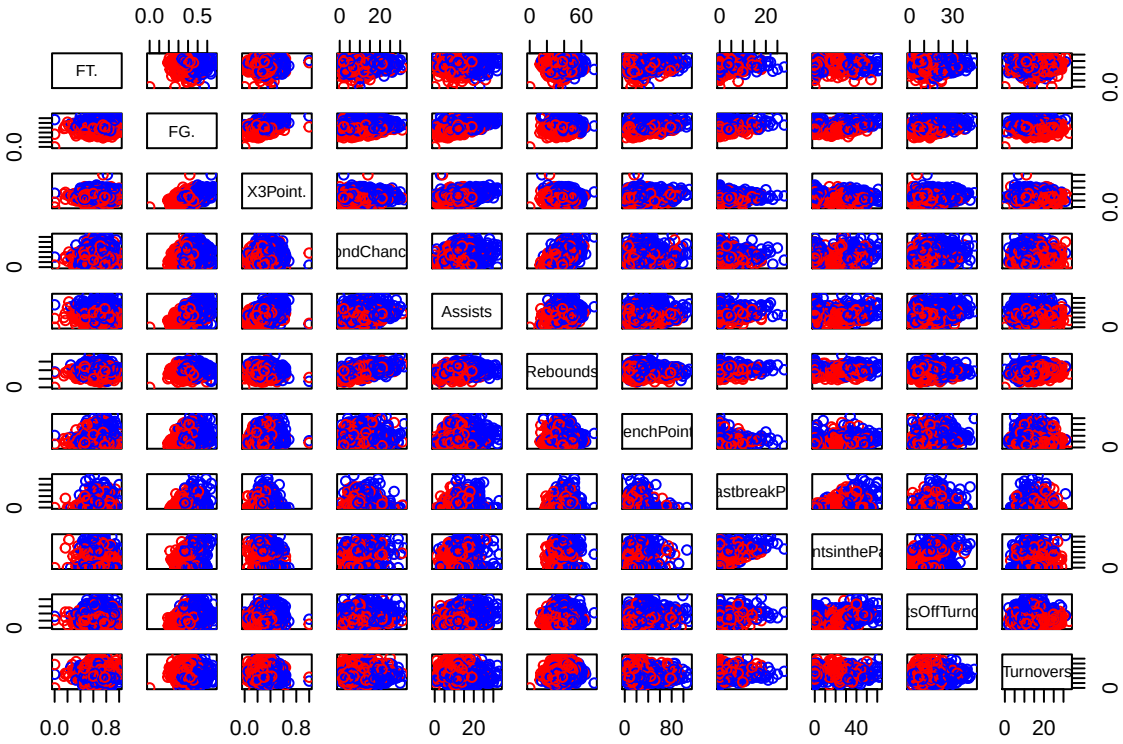
Logistic Regression is a form of regression that is used when the response variable is a binary value (e.g. Success or Failure). In this case we can use our game by game data to create a model that predicts Wins. We previously created a binary variable for Wins and Home, 1 denotes a win, 0 denotes a loss for the Wins variables. And 1 denotes the team is at home, 0 denotes the team as away for the Home variable.

5.2 Scatterplot Matrix

First we'll look at scatterplot matrices of different pairsof features while using colour to distinguish whether a point is a win or a loss.

```
df2_a<-df2
df2_a$Win[df2_a$Win == 1]<-"Win"
df2_a$Win[df2_a$Win == 0]<-"Lose"
cols <- character(nrow(df2_a))
```

```
cols[]<-"black"
cols[df2_a$Win == 'Win'] <- "blue"
cols[df2_a$Win == 'Lose'] <- "red"
pairs(df2_a[,c(4,7,10,13:20)], col = cols)
```



The blue points are wins and the red points are losses.

5.3 Creating the model

We will create it similarly to the second linear regression model but changing the response variable to Wins:

$$Wins = \beta_0 + \beta_1 * FG + \beta_2 * 3P + \beta_3 * FT + \beta_4 * Assists + \beta_5 * Rebounds + \beta_6 * Turnovers$$

```
mod<-glm(Win ~ FG. + X3Point. + FT. +
           Rebounds + Turnovers + Assists,
         binomial(link=logit), data=df2)
summary(mod)
```

```
##
## Call:
## glm(formula = Win ~ FG. + X3Point. + FT. + Rebounds + Turnovers +
##     Assists, family = binomial(link = logit), data = df2)
##
## Deviance Residuals:
```

```
##      Min        1Q    Median        3Q       Max
## -2.97584  -0.64134  -0.00059   0.63748   2.71153
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -15.566277   0.777816 -20.013  < 2e-16 ***
## FG.          21.618184   1.210768  17.855  < 2e-16 ***
## X3Point.      0.633522   0.666129   0.951    0.342
## FT.           2.719948   0.473466   5.745 9.20e-09 ***
## Rebounds      0.153340   0.009189  16.688  < 2e-16 ***
## Turnovers    -0.154919   0.013437 -11.529  < 2e-16 ***
## Assists       0.061059   0.013390   4.560 5.11e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3248.1  on 2342  degrees of freedom
## Residual deviance: 1918.4  on 2336  degrees of freedom
## AIC: 1932.4
##
## Number of Fisher Scoring iterations: 5
```

We can see that all the variables are significant except for 3 Point %.

6 CART

Trees are also very useful techniques. There are two types of trees; Classification And Regression Trees (CART), also known as Decision Trees.

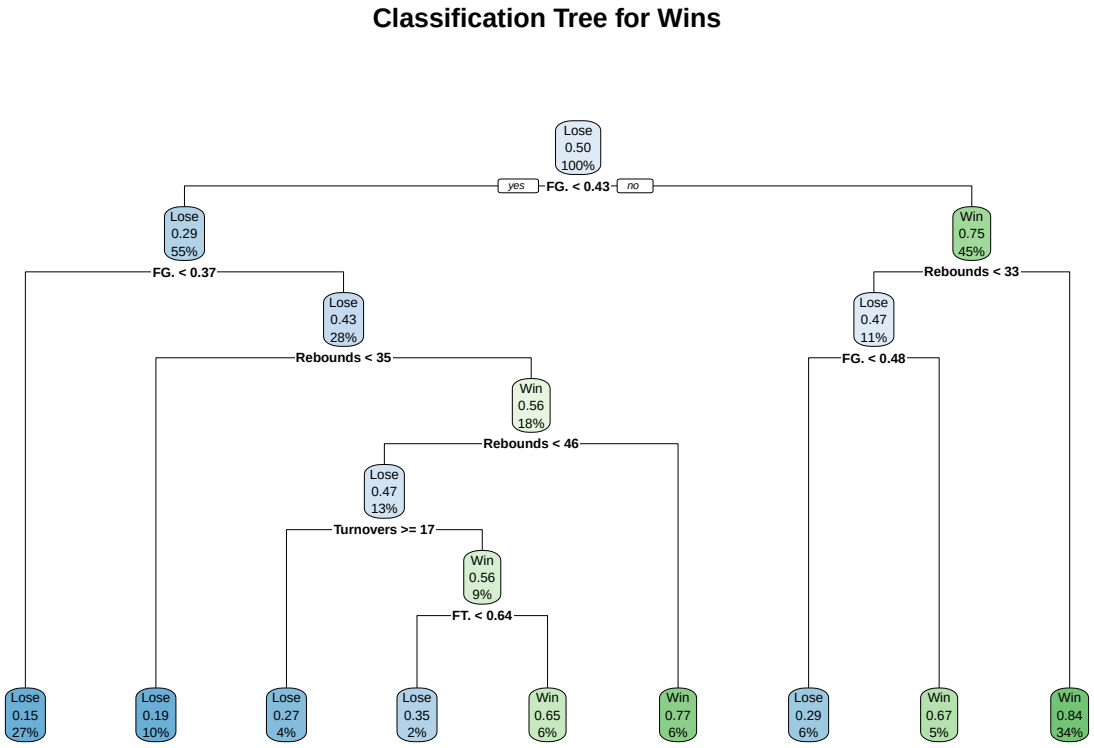
6.1 How they work

The tree takes observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modeling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

Algorithms for constructing decision trees usually work top-down, by choosing a variable at each step that best splits the set of items. Different algorithms use different metrics for measuring "best". These generally measure the homogeneity of the target variable within the subsets.

```
library(rpart)
library(rpart.plot)
dtree<-rpart(Win ~ FT. + FG. + X3Point. + Assists + Rebounds + Turnovers, method = "class", data = df2_a

rpart.plot(dtree, uniform=TRUE,
     main="Classification Tree for Wins")
```



7 Next Steps

Next, I will try to obtain better regression models, try clustering techniques on player data which I'll need to scrape. I also want to try new techniques such as neural networks, SVMs, random forests and possibly deep learning.

8 Appendix

8.1 Data Scraping the OUA website with Python's BeautifulSoup

Obtaining data is a very time consuming and difficult task but with web scraping we can make the task much faster and easier. This is a guide on how I scraped data from the OUA (Ontario University Athletics) website with Python and the Python library, BeautifulSoup. Doing this task requires some Python, and HTML knowledge.

We'll look at the website to get a feel of what data we want and where it is coming from. I was going through this site http://www.oua.ca/sports/mbkb/2018-19/leaders for the men's basketball teams and in the top right you can see a drop down menu where you can select the Season you want to view data from. This is how the webpage looks:

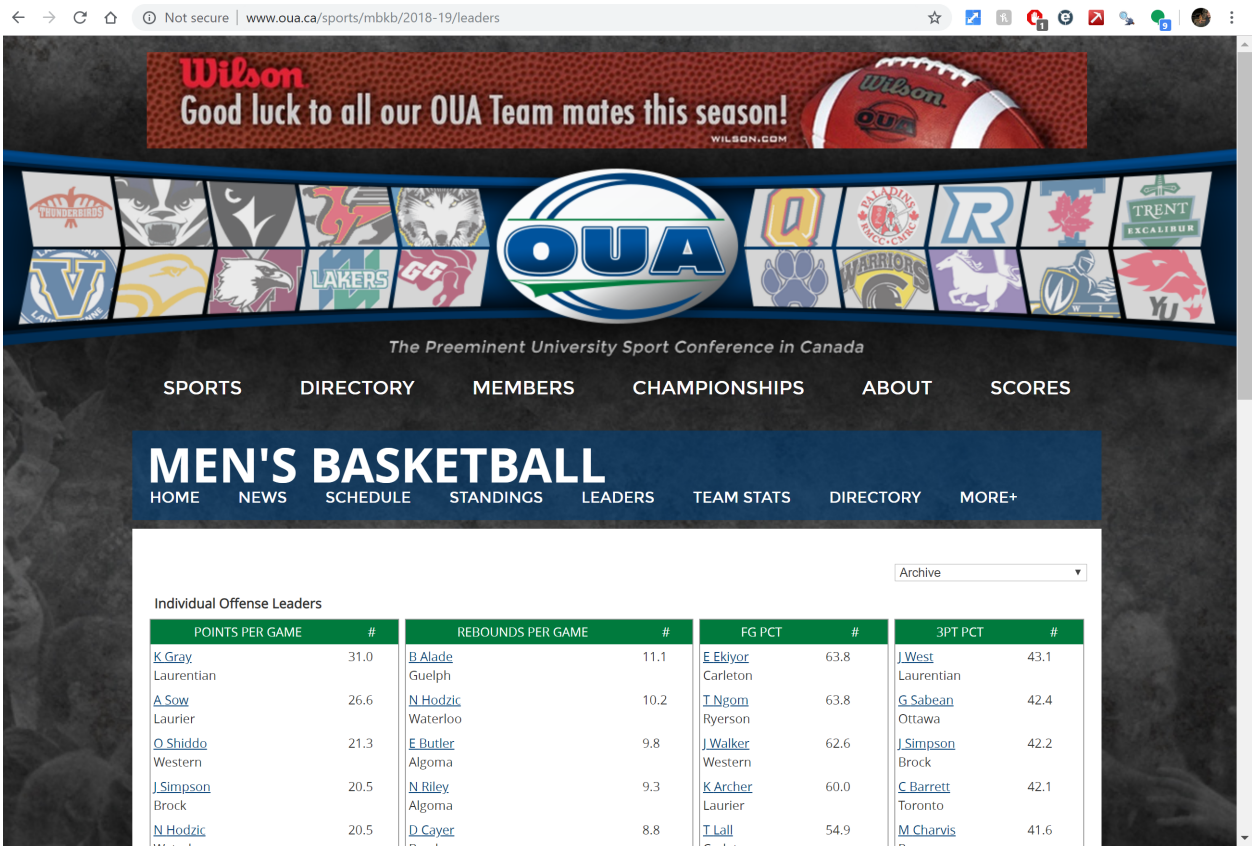


Figure 6: webpage.

After surfing the webpage for desired data, I came across a goldmine. In this table we can select a team and view many different statistics that were collected.

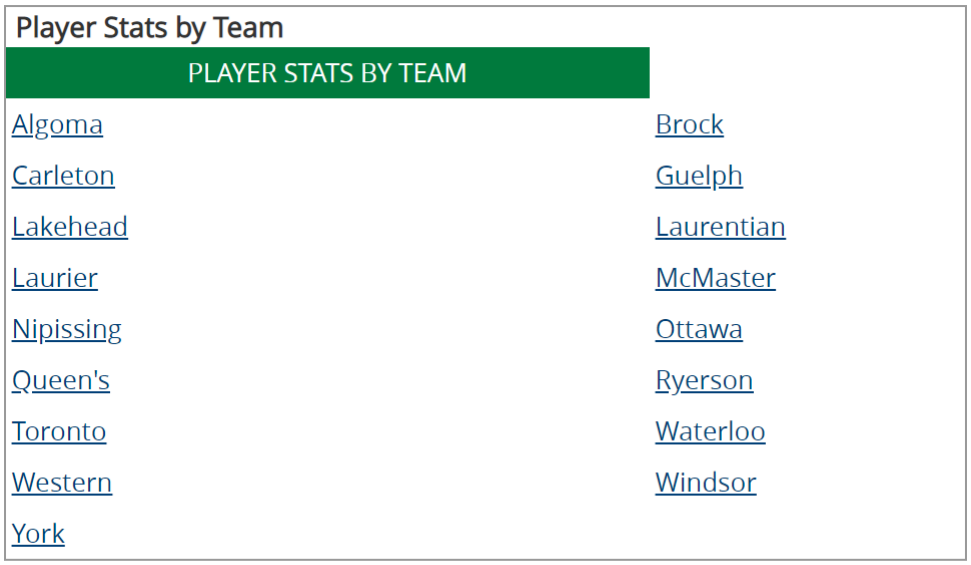
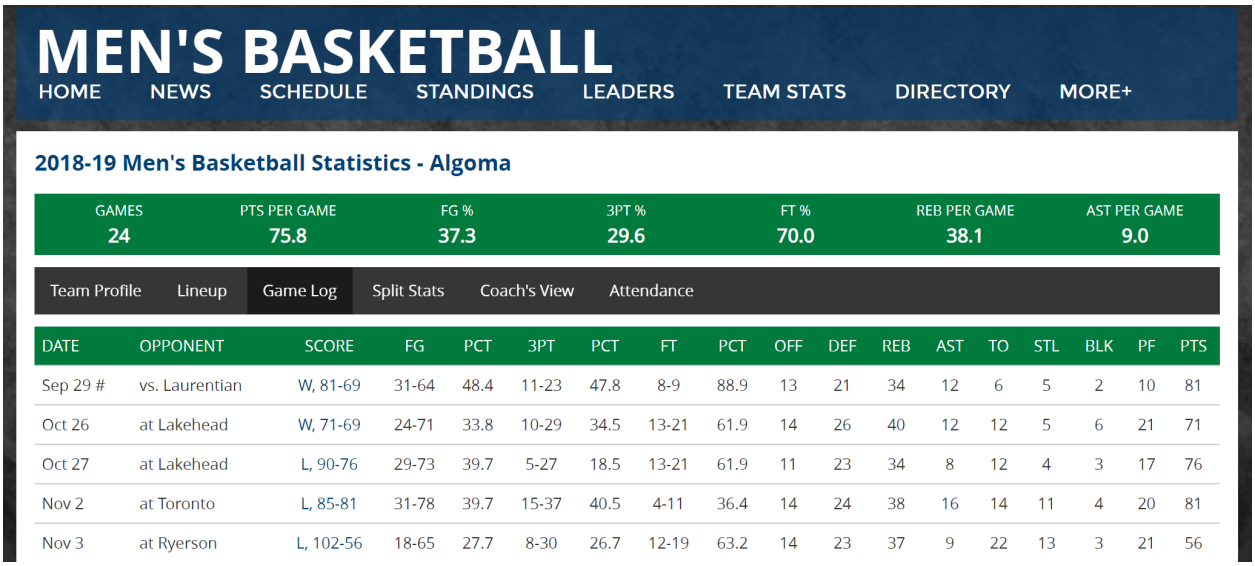


Figure 7: Player Stats by Team.

If we click on a team we can find a game log for the entire regular season when we click on the 'Game Log' tab.



DATE	OPPONENT	SCORE	FG	PCT	3PT	PCT	FT	PCT	OFF	DEF	REB	AST	TO	STL	BLK	PF	PTS
Sep 29 #	vs. Laurentian	W, 81-69	31-64	48.4	11-23	47.8	8-9	88.9	13	21	34	12	6	5	2	10	81
Oct 26	at Lakehead	W, 71-69	24-71	33.8	10-29	34.5	13-21	61.9	14	26	40	12	12	5	6	21	71
Oct 27	at Lakehead	L, 90-76	29-73	39.7	5-27	18.5	13-21	61.9	11	23	34	8	12	4	3	17	76
Nov 2	at Toronto	L, 85-81	31-78	39.7	15-37	40.5	4-11	36.4	14	24	38	16	14	11	4	20	81
Nov 3	at Ryerson	L, 102-56	18-65	27.7	8-30	26.7	12-19	63.2	14	23	37	9	22	13	3	21	56

Figure 8: Gamelog.

From here we can see that each score has an embedded hyperlink and if we click it we can get the game stats. There are player stats of the game, play by play data, team stats, and 1st and 2nd half stats. What I was interested in was the team stats (I'll soon also scrape the player stats and possibly play-by-play data).

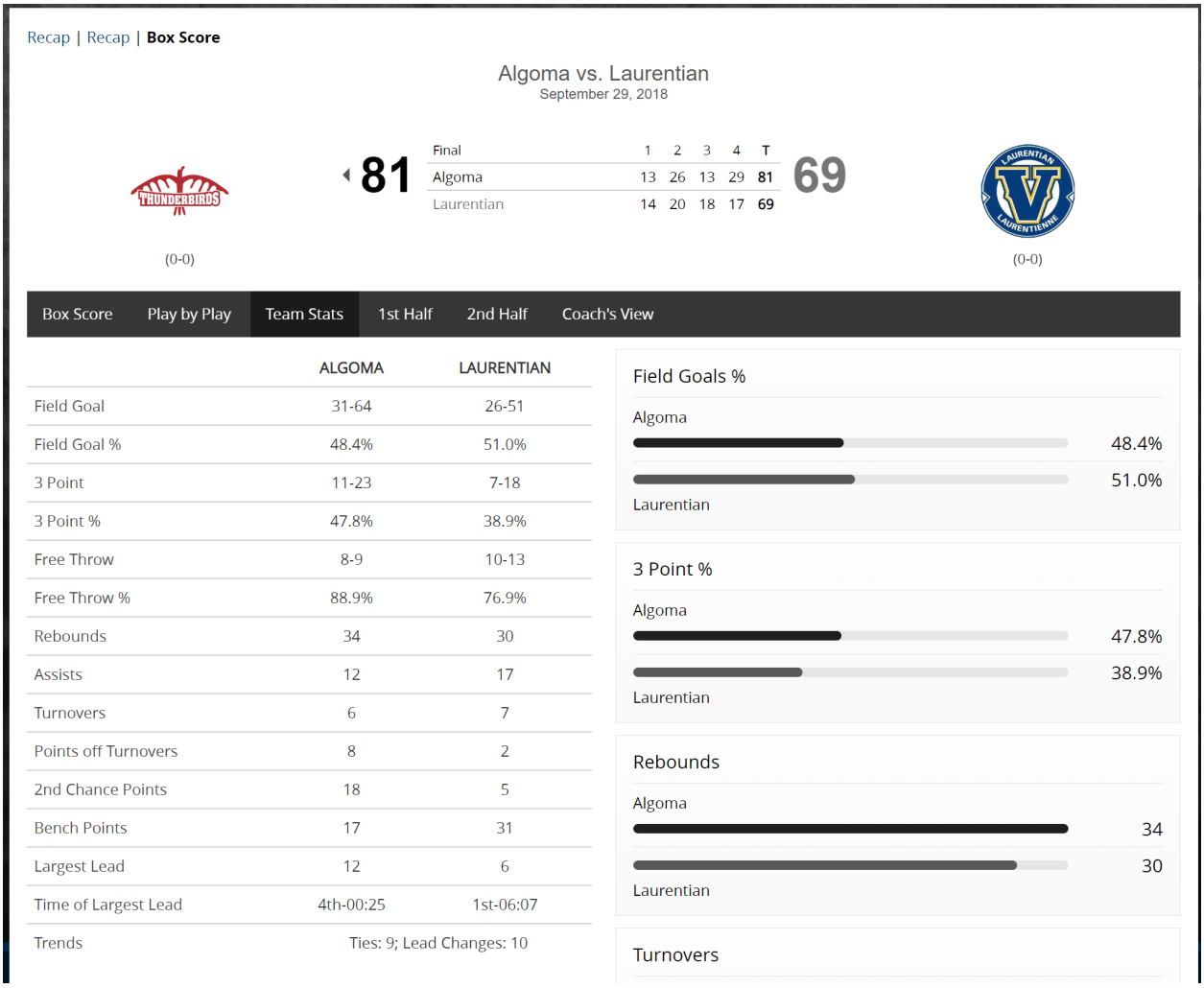


Figure 9: Team Stats.

This is the goldmine right here. We have the game stats for each team and multiple useful tables that we can use. Now this is where the HTML comes in. What we want to do now is inspect elements on the page so that we can see the source code.

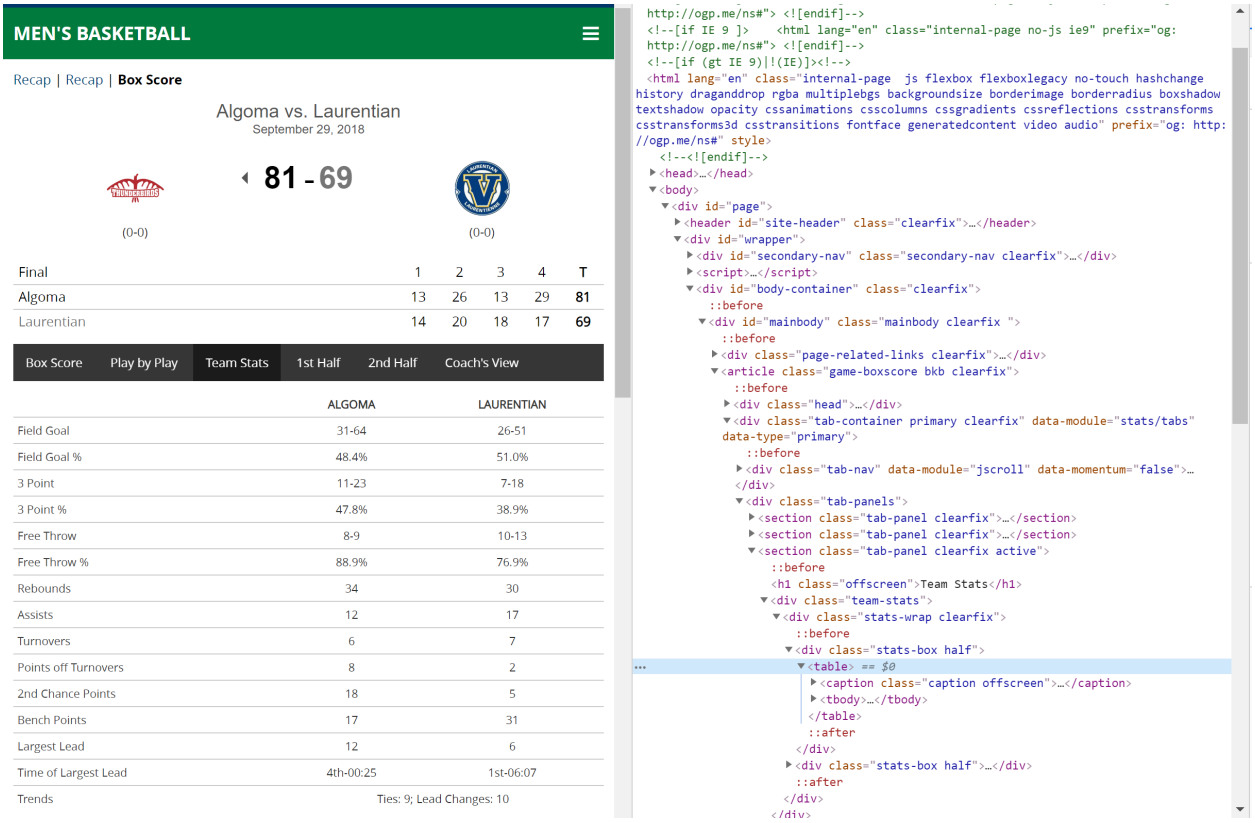


Figure 10: HTML of webpage.

This is where we can see the HTML and its structure. In this page source we can see there are different 'sections'; these are the different tabs (e.g. Box Score tab, Play by Play tab, etc) and in each one there is a table. To scrape the data we need to figure out where the data comes from so that we can reference it and tell Python that we want it. In this table we have a lot of variables like Field Goal for Away and Home team, 3 Point, Turnovers, etc. So we want to find the tags that reference each of them and use some indexing.

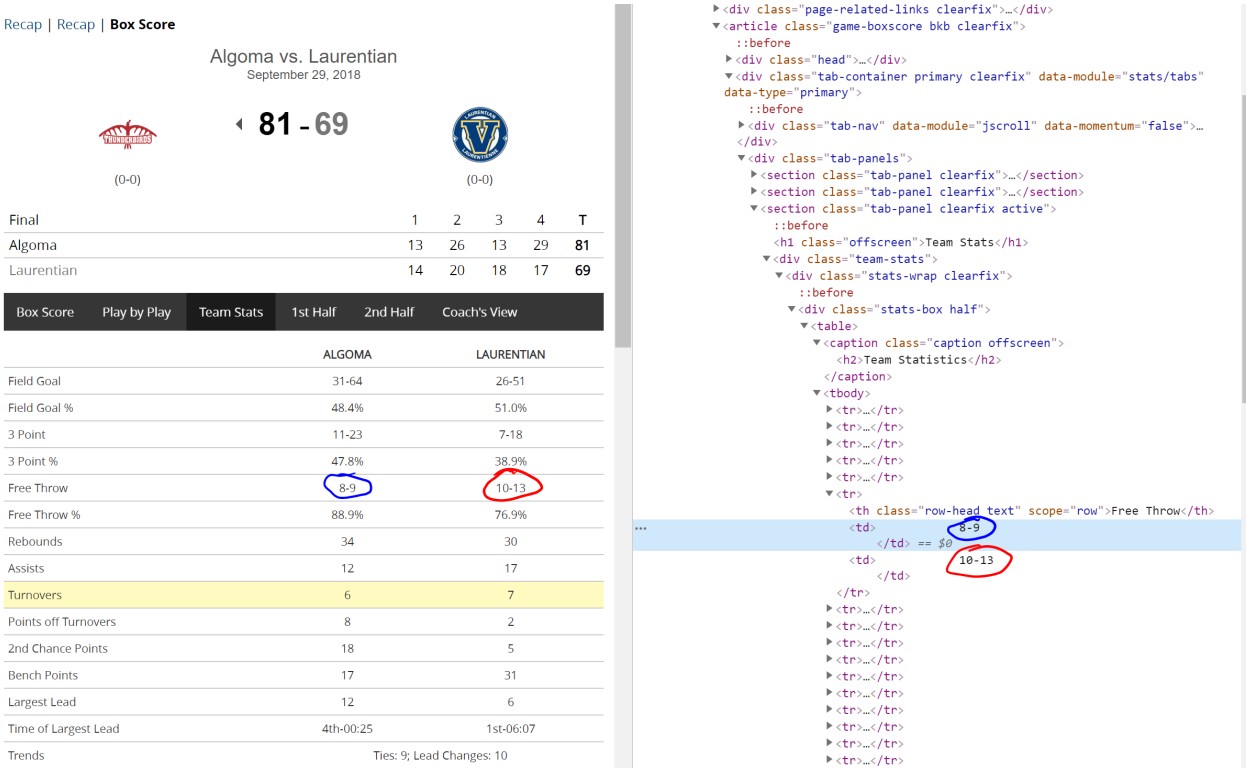


Figure 11: Accessing values using indexing.

Now that we know that the data is shown in the html, we can use Python to find the html tags and collect them.

First we need to start with importing the appropriate library packages.

```
import re
import requests
import pandas as pd
from bs4 import BeautifulSoup
from collections import defaultdict
```

```
r = requests.get('http://www.oua.ca/sports/mbkb/2018-19/boxscores/20180929_svqd.xml?view=teamstats')
#this lets us access the webpage that we want to scrape from
raw_html = r.content
#we can then get the html
soup = BeautifulSoup(raw_html, 'html.parser')
#soup is what we call the html of the webpage which we can use to scrape
stats = soup.findAll("table")
print(stats[8])
```

```
## <table>
## <caption class="caption offscreen"><h2>Team Statistics</h2></caption>
## <tr>
## <th class="col-head" scope="col"><span class="offscreen">Stat</span></th>
## <th class="col-head" scope="col">Algoma</th>
```

```
## <th class="col-head" scope="col">Laurentian</th>
## </tr>
## <tr>
## <th class="row-head text" scope="row">Field Goal</th>
## <td>            31-64
##     </td>
## <td>            26-51
##     </td>
## </tr>
## <tr>
## <th class="row-head text" scope="row">Field Goal %</th>
## <td>            48.4%
##     </td>
## <td>            51.0%
##     </td>
## </tr>
## <tr>
## <th class="row-head text" scope="row">3 Point</th>
## <td>            11-23
##     </td>
## <td>            7-18
##     </td>
## </tr>
## <tr>
## <th class="row-head text" scope="row">3 Point %</th>
## <td>            47.8%
##     </td>
## <td>            38.9%
##     </td>
## </tr>
## <tr>
## <th class="row-head text" scope="row">Free Throw</th>
## <td>            8-9
##     </td>
## <td>            10-13
##     </td>
## </tr>
## <tr>
## <th class="row-head text" scope="row">Free Throw %</th>
## <td>            88.9%
##     </td>
## <td>            76.9%
##     </td>
## </tr>
## <tr>
## <th class="row-head text" scope="row">Rebounds</th>
## <td>34</td>
## <td>30</td>
## </tr>
## <tr>
## <th class="row-head text" scope="row">Assists</th>
## <td>12</td>
## <td>17</td>
## </tr>
```

```
## <tr>
## <th class="row-head text" scope="row">Turnovers</th>
## <td>6</td>
## <td>7</td>
## </tr>
## <tr>
## <th class="row-head text" scope="row">Points off Turnovers</th>
## <td>8</td>
## <td>2</td>
## </tr>
## <tr>
## <th class="row-head text" scope="row">2nd Chance Points</th>
## <td>18</td>
## <td>5</td>
## </tr>
## <tr>
## <th class="row-head text" scope="row">Bench Points</th>
## <td>17</td>
## <td>31</td>
## </tr>
## <tr>
## <th class="row-head text" scope="row">Largest Lead</th>
## <td>12</td>
## <td>6</td>
## </tr>
## <tr>
## <th class="row-head text" scope="row">Time of Largest Lead</th>
## <td>4th-00:25</td>
## <td>1st-06:07</td>
## </tr>
## <tr>
## <th class="row-head text" scope="row">Trends</th>
## <td colspan="2">
##                             Ties: 9;
##                             Lead Changes: 10
##                         </td>
## </tr>
## </table>
```

This is the HTML of the table we want to scrape and we can index different tags to get the desired data.

```
table = stats[8]
print(table.findAll('th', {'scope': 'col'})[1].text.strip())
#This will find the second "th" tag with the scope equal to "col"
#and this prints the Away team
```

```
## Algoma
```

```
print(table.findAll('th', {'scope': 'col'})[2].text.strip())
#This will print the Home team
```

```
## Laurentian
```

This table is good but it doesn't provide the total scores from the game. However we do have the scores at the top of the page.

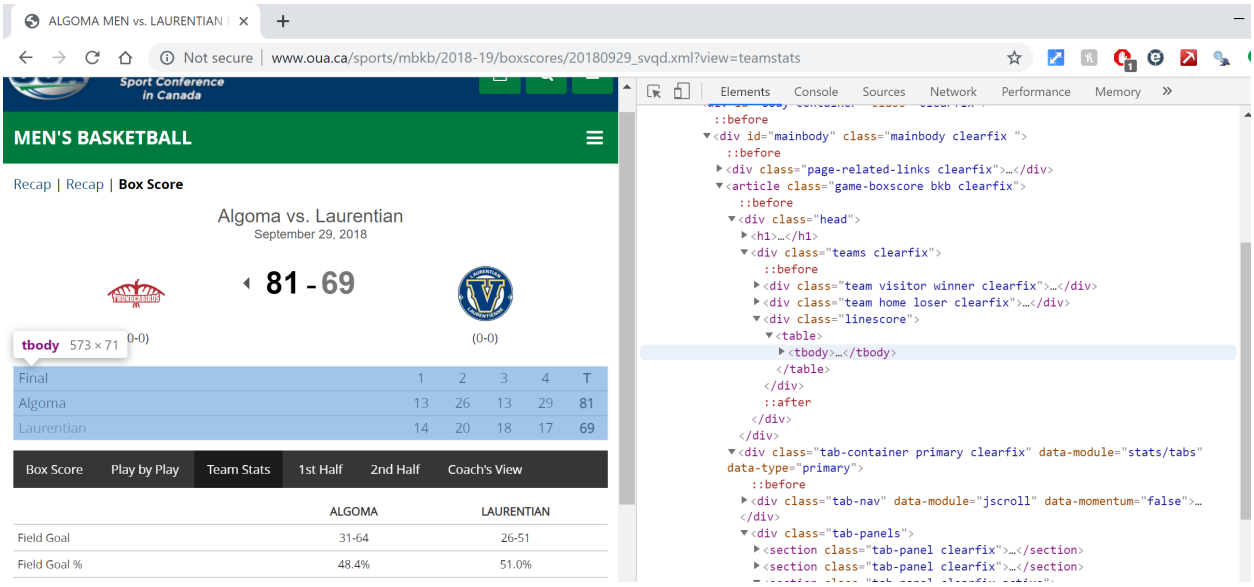


Figure 12: HTML of scores table.

We can use BeautifulSoup to obtain these values as well. After figuring out where all the values are on the webpage, we can create a Python dictionary with keys as our variable names and values as our data that we obtain from the table.

Below is a general function that scrapes the data of an OUA game webpage into a dataframe.

```
def scrape(url):
    """ This function is used to create data dictionaries
    for any url of team stats in the oua website. It
    takes an array of urls but for some games there
    are extra fields to look out for. This function is
    for those that
    do not have those extra fields in the table"""

    # create dictionary for links with less fields in table
    dictlist = {} # initializes a dictionary
    for i in range(len(url)):
    # loop through all the urls in an array
        print(url[i])
        r = requests.get(url[i])
        raw_html = r.content
        soup = BeautifulSoup(raw_html, 'html.parser')
        soup[url[i]] = BeautifulSoup(raw_html, 'html.parser')
        stats = soup[url[i]].findAll("table")
        scores = soup[url[i]].findAll('div', {'class': 'teams clearfix'})[0].table
        # some links have different amounts of tables and sometimes the team
        #stats table is different
        table = stats[8]
        for j in range(2, len(stats)):
```

```
        if str(stats[j].caption) ==
        '<caption class="caption offscreen"><h2>Team Statistics</h2></caption>':
            table=stats[j]
            break
    # this makes sure we get the table we want since it will get the html
    #with the certain caption class (a unique part of the html)

    dictlist[url[i]] = {}
    d = {}
    # we will then create the dictionary with the proper keys and values
    dictlist[url[i]] = {
        "Away" : table.findAll('th', {'scope': 'col'})[1].text.strip(),
        "Home" : table.findAll('th', {'scope': 'col'})[2].text.strip(),

    }
    # some webpages may not have the values listed below so
    #we'll have to watch out for that and prevent getting an index error
    try:
        winner = scores.findAll('tr', {'class': 'winner'})[0]
    except IndexError:
        d[None] = None
    # if the tag is not found we create a null key and value so
    #that it doesn't break
    try:
        loser = scores.findAll('tr', {'class': 'loser'})[0]
    except IndexError:
        d[None] = None
    try:
        dictlist[url[i]].update({"Winner": winner.th.text.strip()})
    except IndexError:
        d[None] = None
    try:
        dictlist[url[i]].update({"Loser": loser.th.text.strip()})
    except IndexError:
        d[None] = None
    try:
        dictlist[url[i]].update({"Winner 1st Qtr Pts":
        winner.findAll('td')[0].text.strip()})
    except IndexError:
        d[None] = None
    try:
        dictlist[url[i]].update({"Loser 1st Qtr Pts":
        loser.findAll('td')[0].text.strip()})
    except IndexError:
        d[None] = None
    try:
        dictlist[url[i]].update({"Winner 2nd Qtr Pts":
        winner.findAll('td')[1].text.strip()})
    except IndexError:
        d[None] = None
    try:
        dictlist[url[i]].update({"Loser 2nd Qtr Pts":
        loser.findAll('td')[1].text.strip()})
```

```
        except IndexError:
            d[None] = None
        try:
            dictlist[url[i]].update({"Winner 3rd Qtr Pts":
            winner.findAll('td')[2].text.strip()})
        except IndexError:
            d[None] = None
        try:
            dictlist[url[i]].update({"Loser 3rd Qtr Pts":
            loser.findAll('td')[2].text.strip()})
        except IndexError:
            d[None] = None
        try:
            dictlist[url[i]].update({"Winner 4th Qtr Pts":
            winner.findAll('td')[3].text.strip()})
        except IndexError:
            d[None] = None
        try:
            dictlist[url[i]].update({"Loser 4th Qtr Pts":
            loser.findAll('td')[3].text.strip()})
        except IndexError:
            d[None] = None
        try:
            dictlist[url[i]].update( {"Winner Total Pts":
            winner.findAll('td')[4].text.strip()})
        except IndexError:
            d[None] = None
        try:
            dictlist[url[i]].update( {"Loser Total Pts":
            loser.findAll('td')[4].text.strip()})
        except IndexError:
            d[None] = None
        for j in range(16):
            try:
                dictlist[url[i]].update( { table.findAll('th', {'scope':
                'row'})[j].text.strip() + ' Away':
                table.findAll('td')[2*j].text.strip()})
            except IndexError:
                d[None] = None
            try:
                dictlist[url[i]].update({ table.findAll('th', {'scope':
                'row'})[j].text.strip() + ' Home' :
                table.findAll('td')[2*j+1].text.strip()})
            except IndexError:
                d[None] = None
            try:
                dictlist[url[i]].update({table.findAll('th', {'scope':
                'row'})[16].text.strip()+' Away':
                table.findAll('td')[32].text.strip()})
            except IndexError:
                d[None] = None
    z = {**dictlist, **d}
    # we then merge the two dictionaries so that it shows
```

```
    #which values are missing

    return z
```

Using this function we can get the data from any array of game websites and it is generalized to prevent errors and obtain the right fields for the associated keys.

Note: Some games can go to overtime but I found that an easy fix on Excel. If you sort the total points by least to greatest we can see that the quarters won't add up to the total so I created another column for the winner and losers and summed the entries to get the actual total.

Collecting all the links would be time consuming as well so a function was made for that as well. We can use BeautifulSoup for this too since the html for the game log page contains a tag called href which is the url for the webpages that we are interested in.

```
def get_links():
    """ All links have a certain pattern since they are all similar.
    However there is a part of the url that makes
     it unique and so we obtain the links from the html """
    print("getting links...")
    teams = ['algoma', 'brock', 'carleton', 'guelph',
    'lakehead', 'laurentian',
             'laurier', 'mcmaster', 'nipissing',
             'ottawa', 'queens', 'ryerson',
             'toronto', 'waterloo', 'western',
             'windsor', 'york']
    years = ['2014-15', '2015-16', '2016-17', '2017-18', '2018-19']
    original_url = 'http://oua.ca/sports/mbkb/'
    end_url = '?view=gamelog'
    href_list = []
    for year in years:
        for team in teams:
            current_url = original_url + year + '/teams/' + team + end_url
            r = requests.get(current_url)
            raw_html = r.content
            soup = BeautifulSoup(raw_html, 'html.parser')
            tables = soup.findAll('table')
            max_len = 0
            index = 0

            for i in range(len(tables)):
                tags = tables[i].findAll('a')
                if len(tags) > 0:
                    url = tags[0].get('href', None)
                    if "/boxscores/20" in url and len(tables[i]) > max_len:
                        index = i
                        max_len = len(tables[i])

            table = tables[index]

            tags = table.findAll('a')
            for tag in tags:
                url = re.sub("\.\.", original_url + year, tag.get('href', None))
                url += '?view=teamstats'
```

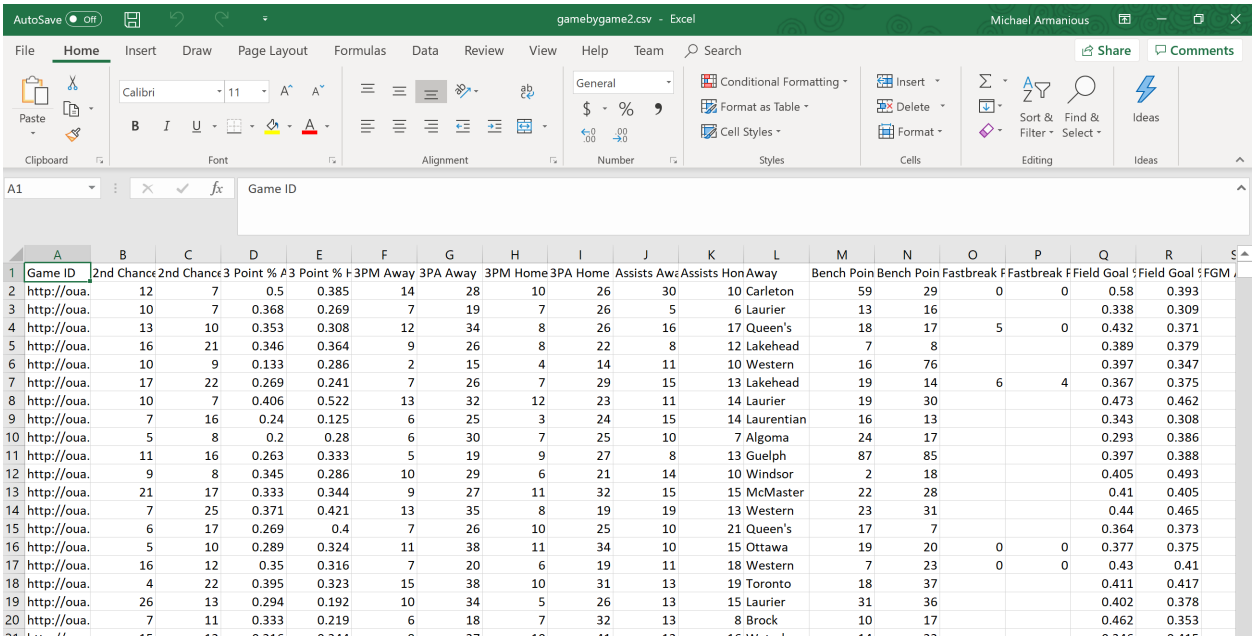
```
                href_list.append(url)

    print("done getting links")
    return href_list
```

Using both these functions together we can obtain our data and put it into a dataframe.

```
n [ ]:
    q = get_links()
    a = scrape(q)
    df = pd.DataFrame(a)
    df = df.T
    df = df.replace('\-', ' -- ', regex=True).astype(object)
    #This is used because when opening in Excel
    #it automatically assumes that the FG is a date (e.g. FG: 10-12)
    df = df[['Away', 'FG Away', 'FG% Away', '3PT FG Away',
    '3PT FG% Away', 'FT Away', 'FT% Away', 'Rebounds Away',
             'Assists Away',
             'Turnovers Away', 'Points Off Turnovers Away',
             '2nd Chance Points Away', 'Points in the Paint Away',
             'Fastbreak Points Away', 'Bench Points Away',
             'Largest Lead Away', 'Time of Largest Lead Away',
             'Home', 'FG Home',
             'FG% Home', '3PT FG Home', '3PT FG% Home',
             'FT Home', 'FT% Home', 'Rebounds Home', 'Assists Home',
             'Turnovers Home',
             'Points Off Turnovers Home', '2nd Chance Points Home',
             'Points in the Paint Home', 'Fastbreak Points Home',
             'Bench Points Home', 'Largest Lead Away',
             'Time of Largest Lead Away', 'Trends Away', 'Winner',
             'Winner 1st Qtr Pts',
             'Winner 2nd Qtr Pts', 'Winner 3rd Qtr Pts',
             'Winner 4th Qtr Pts', 'Winner Total Pts', 'Loser',
             'Loser 1st Qtr Pts', 'Loser 2nd Qtr Pts',
             'Loser 3rd Qtr Pts', 'Loser 4th Qtr Pts', 'Loser Total Pts']]

    df.to_csv('gamebygame2.csv', header=True)
```



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Game ID	2nd Chance	2nd Chance	3 Point % A	3 Point % H	3PM Away	3PA Away	3PM Home	3PA Home	Assists Awa	Assists Hon	Away	Bench Poin	Bench Poin	Fastbreak F	Fastbreak F	Field Goal %	Field Goal %	FGM
2	http://oua.	12	7	0.5	0.385	14	28	10	26	30	10	Carleton	59	29	0	0	0.58	0.393	
3	http://oua.	10	7	0.368	0.269	7	19	7	26	5	6	Laurier	13	16			0.338	0.309	
4	http://oua.	13	10	0.353	0.308	12	34	8	26	16	17	Queen's	18	17	5	0	0.432	0.371	
5	http://oua.	16	21	0.346	0.364	9	26	8	22	8	12	Lakehead	7	8			0.389	0.379	
6	http://oua.	10	9	0.133	0.286	2	15	4	14	11	10	Western	16	76			0.397	0.347	
7	http://oua.	17	22	0.269	0.241	7	26	7	29	15	13	Lakehead	19	14	6	4	0.367	0.375	
8	http://oua.	10	7	0.406	0.522	13	32	12	23	11	14	Laurier	19	30			0.473	0.462	
9	http://oua.	7	16	0.24	0.125	6	25	3	24	15	14	Laurentian	16	13			0.343	0.308	
10	http://oua.	5	8	0.2	0.28	6	30	7	25	10	7	Algoma	24	17			0.293	0.386	
11	http://oua.	11	16	0.263	0.333	5	19	9	27	8	13	Guelph	87	85			0.397	0.388	
12	http://oua.	9	8	0.345	0.286	10	29	6	21	14	10	Windsor	2	18			0.405	0.493	
13	http://oua.	21	17	0.333	0.344	9	27	11	32	15	15	McMaster	22	28			0.41	0.405	
14	http://oua.	7	25	0.371	0.421	13	35	8	19	19	13	Western	23	31			0.44	0.465	
15	http://oua.	6	17	0.269	0.4	7	26	10	25	10	21	Queen's	17	7			0.364	0.373	
16	http://oua.	5	10	0.289	0.324	11	38	11	34	10	15	Ottawa	19	20	0	0	0.377	0.375	
17	http://oua.	16	12	0.35	0.316	7	20	6	19	11	18	Western	7	23	0	0	0.43	0.41	
18	http://oua.	4	22	0.395	0.323	15	38	10	31	13	19	Toronto	18	37			0.411	0.417	
19	http://oua.	26	13	0.294	0.192	10	34	5	26	13	15	Laurier	31	36			0.402	0.378	
20	http://oua.	7	11	0.333	0.219	6	18	7	32	13	8	Brock	10	17			0.462	0.353	

Figure 13: Final Datafram.